Intégrateurs exponentiels modifiés pour la simulation des vagues non linéaires

Brice Eichwald

Université Nice Sophia Antipolis - U.F.R. Sciences

école Doctorale Sciences Fondamentales et Appliquées n°364

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université Nice Sophia Antipolis (UNS)

Spécialité : Physique

présentée et soutenue par

M. Brice Eichwald

Intégrateurs exponentiels modifiés pour la simulation des vagues non linéaires

---

Thèse dirigée par M. Didier Clamond et M. Marc Francius

soutenue le 05/07/2013

Jury :

Rapporteur	M. Malek Abid	MCF	Universités d'Aix-Marseille I & II
Rapporteur	M. Michel Benoit	PR	Laboratoire d'Hydraulique Saint-Venant
Directeur	M. Didier Clamond	PR	Université Nice Sophia Antipolis
Co-Directeur	M. Marc Francius	MCF	Université du Sud-Toulon-Var
Examinateur	M. Denys Dutykh	CR	CNRS - Université de Savoie

Remerciements

Tout d'abord je souhaite remercier mes deux directeurs pour m'avoir accordé leur confiance et avoir encadré cette thèse ainsi que les membres du Jury pour leurs retours sur ce manuscrit et leur présence à ma soutenance.

Je remercie mes différents collègues de bureau pour la très bonne ambiance de travail créée durant ces années de doctorat, sans oublier tous ceux avec lesquels j'ai passé d'excellents moments ni ceux avec lesquels j'ai eu de nombreux échanges techniques et scientifiques. Enfin, je fais une *spéciale dédicace* à un ami de longue date, Maxence, avec lequel j'ai eu des discussions scientifiques fructueuses.

Cette thèse est l'aboutissement de 9 années passées au sein de l'Université Nice Sophia Antipolis, plus exactement à la Faculté des Sciences du Campus Valrose, et j'ai une pensée, en ce jour, pour tous ceux qui m'ont accompagné depuis ma première année de Licence : étudiants, enseignants et personnels administratifs et techniques.

Table des matières

Introduction

Objectifs de recherches

Lorsque nous essayons de résoudre un modèle scientifique de manière numérique, nous faisons appel à une avance temporelle pour calculer la solution d'une équation d'évolution. Le choix de cette avance se fait, en général, selon des critères de précision recherchée et de temps de calcul. Or, un inconvénient majeur des méthodes numériques est que ce temps nécessaire à la réalisation de simulations numériques croît rapidement avec la difficulté de calcul, même pour un ordinateur puissant. Ainsi, pour obtenir des résultats de simulations dans des temps raisonnables, avec, par exemple, des équations fortement non linéaires, il existe deux approches possibles. Soit prendre en compte une tolérance aux erreurs numériques élevée, ce qui a pour conséquence de donner des résultats rapides mais peu fiables, soit utiliser des méthodes numériques de hautes performances qui permettent *d'optimiser* le temps de calcul à des tolérances faibles. Ces dernières nous permettent de *capter* la physique sous-jacente et donc les événements rares qui sont souvent les plus importants, mais elles sont aussi généralement fastidieuses à comprendre et à mettre au point dans des codes numériques déjà existants.

Les difficultés de calculs sont issues de ce qui est nommée la *raideur numérique*, ou « stiffness ». Il n'existe pas de définition bien précise de ce phénomène. La première définition, qui est aussi selon Hairer *et al.* [42] la plus pragmatique, est la suivante : les problèmes *raides numériquement* sont ceux pour lesquels les méthodes temporelles *implicites* sont beaucoup plus performantes que les méthodes temporelles *explicites*, à cause de pas de temps d'évolution temporelle beaucoup trop petits. De manière générale, cela apparaît dans un système contenant plusieurs temps caractéristiques et provient principalement des termes linéaires de l'équation d'évolution. Mais, par exemple, dans le cas des équations de vagues, il s'avère que plus l'onde considérée est grande ou pentue, plus il est difficile d'obtenir une approximation numérique de la solution en un temps raisonnable. En effet, lors de rapides variations dans le profil de la vague, les termes qui doivent être calculés deviennent de plus en plus difficiles à manipuler. Il est cohérent que cela se produise de temps en temps lorsque nous avons ponctuellement une forte variation du profil de l'onde (par exemple lors d'interactions ou de perturbations), mais d'après les problèmes étudiés, cela arrive aussi lorsque le système considéré ne présente aucune difficulté apparente.

Nous souhaitons donc mettre au point une nouvelle approche qui offrirait plusieurs avantages

par rapport à celles qui existent déjà, à savoir :
- Facilité de compréhension ;
- Augmentation de la taille des pas de temps pour l'évolution temporelle ;
- Conservation des ordres de grandeur des erreurs numériques ;
- Implémentation aisée sans devoir recoder un code existant en entier.

En observant que toutes les équations modèles considérées ont comme point commun de pouvoir se mettre sous la forme d'équations aux dérivées ordinaires, nous allons travailler en partant de ce constat. Pour atteindre notre objectif, nous modifions une méthode existante, faisant partie de la classe des méthodes dites d'*intégrateurs exponentiels*, le *Facteur Intégrant* classique. Cette méthode permet de retirer le terme linéaire de l'équation d'évolution, grâce à un certain changement de variable. Ainsi, nous enlevons la raideur numérique issue de ce terme. Notre but est d'aller encore plus loin, en essayant de faire de même, mais cette fois sur la partie non linéaire, afin de réduire au maximum la difficulté de calcul. Nous obtenons ainsi notre *Facteur Intégrant Modifié*.

Application au modèle Proies et Prédateurs

La première étape de notre réflexion a été de tester notre facteur intégrant modifié avec des équations *très simples* à manipuler, afin de voir s'il était utile de se lancer dans un développement plus poussé. Notre choix s'est porté sur un modèle couramment utilisé pour décrire la dynamique de systèmes biologiques, dans lesquels un prédateur et sa proie interagissent : celui des *Proies et Prédateurs*, constitué du couple d'équations différentielles non linéaires que nous devons à Alfred James Lotka et Vito Volterra [54, 79]. De plus, ce modèle est aussi utilisé en économie sous le nom de *modèle Goodwin* [37].
Les prédateurs et les proies, usuellement notés u et v, sont liés par le système d'équations

$$u_t = u\left(\alpha - \beta v\right), \qquad v_t = -v\left(\delta - \gamma u\right) \tag{1}$$

Nous faisons le choix du jeu de paramètres $\alpha = \beta = -\dfrac{1}{2}$ et $\delta = \gamma = -1$ et de la condition initiale $u\left(0\right) = 2$ et $v\left(0\right) = 1$. Cela nous donne le système

$$u_t = \frac{u}{2}\left(v - 1\right), \qquad v_t = v\left(1 - u\right) \tag{2}$$

Nous pouvons mettre les équations précédentes sous la forme suivante, en séparant parties linéaire et non linéaire, avec la matrice des parties linéaires $\mathbb{A} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & -1 \end{pmatrix}$,

$$\frac{\partial}{\partial t}\begin{pmatrix} u \\ v \end{pmatrix} + \mathbb{A}\cdot\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{uv}{2} \\ -uv \end{pmatrix} \tag{3}$$

Lorsque nous comparons le nombre de pas de temps nécessaires pour réaliser une même simulation avec différentes méthodes, nous constatons d'importantes différences comme sur la figure 1 représentant l'évolution de ces pas de temps.

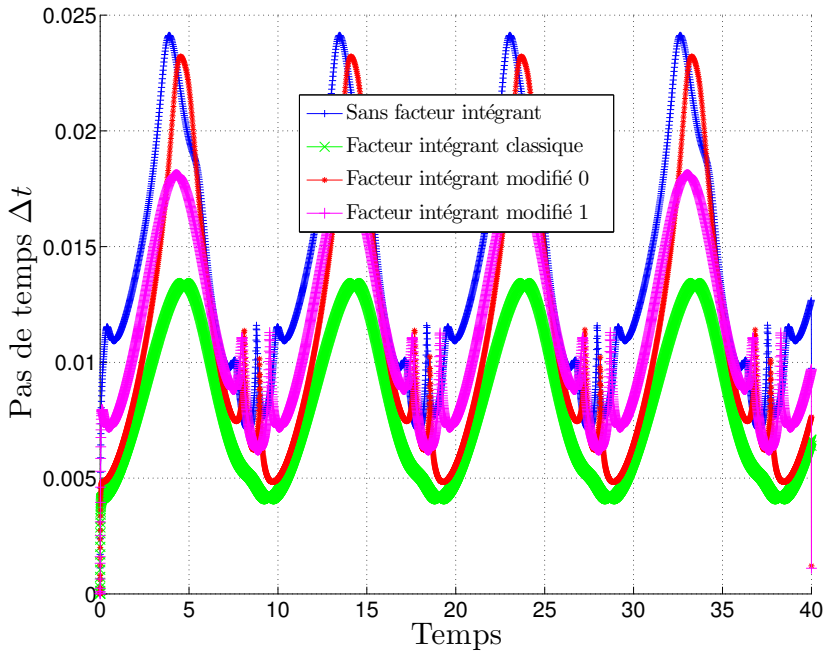

FIGURE 1 – Evolution du pas de temps Δt en fonction du temps de simulation pour différents schémas numériques.

En effet, pour une méthode d'avancement temporelle classique, sans aucun facteur intégrant (en bleu), il nous faut 3 275 pas de temps pour réaliser une simulation test, contre 5 789 pour une méthode avec le facteur intégrant classique. Donc cette dernière est moins intéressante que la méthode temporelle classique, puisqu'il faut près du double de pas de temps pour conclure la simulation. Par contre, en prenant à la place du facteur intégrant classique notre facteur intégrant modifié aux différents ordres 0 et 1, nous nous apercevons que les pas de temps sont plus grands et que cela va en s'améliorant avec l'ordre de la méthode. En effet, avec notre facteur intégrant modifié, nous sommes capables de fortement réduire le nombre total de pas de temps nécessaires à la simulation : 4 538 et 3 900. Au bout d'un temps très long de simulation et pour toutes les méthodes citées, les quantités de proies et prédateurs ne diffèrent que de l'ordre de la tolérance choisie sur les erreurs numériques. Il n'est pas incohérent de voir que la méthode d'avancement temporelle seule est plus rapide que celle avec le facteur intégrant classique, puisque la méthode temporelle choisie résout mieux les solutions de forme polynomiales qu'exponentielles. Or, lors du changement de variable pour utiliser le facteur intégrant classique, nous changeons la forme de la solution intermédiaire en faisant intervenir une certaine exponentielle de matrice dans le calcul, ce qui rend l'évolution temporelle moins aisée. Néanmoins, lorsque le facteur intégrant classique rajoute une sorte de complexité, ou raideur numérique, qui rend le calcul plus difficile, le facteur intégrant modifié nous montre sa supériorité. Cela s'explique par le fait que nous pouvons, avec notre nouvelle méthode, rattraper le défaut du facteur intégrant classique, en réduisant la difficulté de calcul ajoutée. Ce premier test nous révèle donc un potentiel intéressant pour notre nouvelle approche.

Choix de l'étude liée aux vagues

Une fois que nous avons vu que notre facteur intégrant modifié semblait performant, il a fallu choisir un cadre d'étude plus poussé. Comme j'ai réalisé mes stages de Master sur des thématiques liées à l'eau et que ma formation de physicien m'a sensibilisé à l'étude de

phénomènes physiques *de tous les jours*, le choix des vagues s'est imposé de lui-même. En effet, en milieu marin, les éléments naturels qui semblent les plus évidents à étudier du fait de leurs omniprésence à la surface de l'eau sont les vagues, pour lesquelles il existe de nombreux modèles, soit de petites vagues (en faible profondeur) ou soit des vagues dites *extrêmes* (les tsunamis, les marées de tempêtes ou les vagues scélérates). Les vagues qui posent le plus de problèmes à la société sont celles qui sont extrêmes. En effet, lorsque la mer se déchaîne, les dégâts qu'elles causent sont souvent très coûteux en terme de matériels et malheureusement aussi en terme de vies humaines. Si les premiers cités sont, entre autres, désagréables d'un point de vue financier, que dire des seconds ... Un exemple frappant reste celui du bateau de croisière *MV Louis Majesty* qui naviguait au large de l'Espagne, au milieu d'une mer agitée mais sans aucun avis de tempête prévu. Le 4 mars 2010, quelques mois à peine après le début de cette thèse, trois vagues extrêmes de 8 mètres de haut ont frappé le navire et plus précisément le pont supérieur, en brisant les baies vitrées d'un salon et permettant ainsi aux vagues suivantes de s'engouffrer. Ces vagues dites *scélérates* (pouvant atteindre jusqu'à 30 mètres de haut) ont fait 2 morts et 14 blessés. Afin de mieux comprendre et prédire ce type de vagues dévastatrices, il faut développer des modèles scientifiques complexes. Or, cela s'accompagne souvent de difficultés de calcul et de raideur numérique importante. Selon la méthode utilisée, il est donc possible de *passer à côté* des événements rares que sont ces vagues extrêmes. Nous pensons donc que si nous pouvons utiliser notre facteur intégrant modifié pour *accélérer* l'étude de vagues, il nous sera possible d'obtenir des avancées significatives à la vue des résultats du modèle précédent de proies et prédateurs.

Plan de ce rapport de Thèse

Avant de rentrer dans le vif du sujet, nous donnons des rappels généraux sur différentes thématiques, comme sur les équations des vagues considérées ici (chapitre I) ou sur les méthodes spectrales et les différentes méthodes numériques d'avancement temporel utilisées (chapitre II).

Nous faisons un tour d'horizon des méthodes d'*intégrateurs exponentiels*, comprenant le *Facteur Intégrant* classique, dans le chapitre III, suivi de notre *Facteur Intégrant Modifié* au chapitre IV.

Afin d'étudier notre approche, nous l'appliquons à différents modèles de vagues plus ou moins complexes et riches en dynamique. Nous commençons par des équations assez *simples* à simuler. Ce sont celles de *Korteweg et de Vries* et de *Benjamin, Bona et Mahony* qui font appel à de nombreuses approximations, ainsi que celle de *Schrödinger Non Linéaire* (chapitre V). Ces trois modèles nous permettent de valider l'implémentation des différentes méthodes utilisées avant d'étudier plus en détails le cas de systèmes d'équations couplées : les modèles de *Serre* (chapitre VI) contenant moins d'approximations et de *High-Order Spectral* (chapitre VII), modèle le plus fortement non linéaire à notre disposition et qui est obtenu sans faire d'approximations. Le choix de ces équations des vagues est fait pour augmenter la difficulté de résolution des termes non linéaires par le solveur utilisé, ce qui implique la nécessité d'en améliorer l'efficacité. Avec ces différents tests numériques, nous pouvons avoir une idée précise de l'intérêt de la méthode que nous développons.

Ces différentes équations de vagues nous permettent de simuler l'évolution de la surface libre de l'eau. Il est aussi tout à fait envisageable d'appliquer notre travail aux équations traitant de l'interface entre deux fluides (pour rester dans le cadre de l'hydrodynamique), ou encore à l'étude d'un plasma. Mais surtout, ce travail pourra être utilisé pour d'autres champs de recherches, si nous avons à notre disposition des équations aux dérivées ordinaires.

Chapitre I

Equations des vagues

Il existe deux grandes théories pour modéliser la dynamique des vagues :

- La *théorie des ondes longues*, pour laquelle la période de l'onde est beaucoup plus grande que la profondeur.
- La *théorie en profondeur infinie*, pour laquelle la période de l'onde est beaucoup plus petite que la profondeur.

Nous introduisons le profil des vagues en I.1 ainsi que les différentes équations étudiées en I.2.

1 Le profil des ondes progressives

De manière générale, une vague progressive est définie par le profil schématisé sur la figure I.1 et possède :

- Une amplitude de la crête à la hauteur moyenne a ;
- Une amplitude du creux à la hauteur moyenne b ;
- Une amplitude crête à creux $H = a + b$;
- Une profondeur d'eau sous le niveau moyen d ;
- Une élévation de la surface libre par rapport au niveau moyen η ;
- Une période λ.

Le système d'écoulement à la surface libre étudié ici est soumis à l'accélération de la gravité g, l'axe des x correspondant à la direction de propagation horizontale et l'axe des z à la direction verticale (perpendiculaire à la surface au repos).

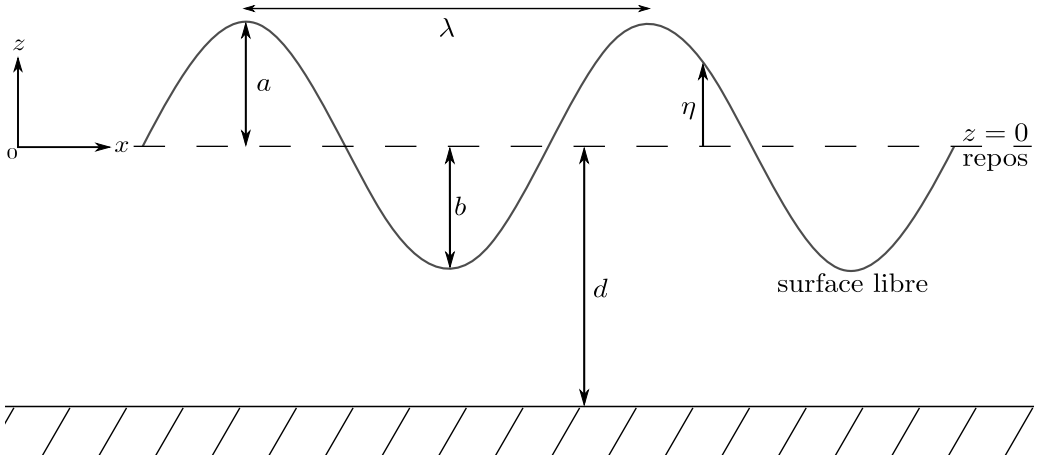

FIGURE I.1 – Schéma du système étudié tout au long de ce manuscrit.

Onde solitaire

En 1834, J. S. Russell [62] qui se promenait le long de l'Union canal, reliant Edimbourg à Forth-Clyden en Ecosse, a observé une onde solitaire de forte amplitude générée par l'arrêt brusque d'une barge. Ce type d'onde qui se propage sur une longue distance et sans aucune déformation est une onde solitaire particulière. Elle possède une énergie spatialement localisée et est extrêmement stable vis-à-vis de certaines perturbations.
Cette onde est solution de nombreuses équations aux dérivées partielles non linéaires, ce qui fait que nous l'observons dans de nombreux phénomènes physiques, allant des vagues à l'optique. La première théorie est à créditer à Joseph Valentin Boussinesq [9].

Onde cnoïdale

Pour les équations des vagues considérées ici, il existe des solutions périodiques de forme permanente en eau peu profonde, les *ondes cnoïdales* [48], dont nous voyons une photo à la figure I.2. L'onde solitaire précédente n'est qu'un cas particulier des ondes cnoïdales. Il est même possible, à l'aide d'une renormalisation, d'utiliser ces dernières pour des vagues en eau profonde, voire même en profondeur infinie [19, 20]. Un avantage de ces ondes est que nous pouvons faire varier leur profil à l'aide d'un seul paramètre. Cela nous permet de tester un modèle d'équation de vagues avec une solution exacte, ayant la possibilité de prendre des profils assez différents.

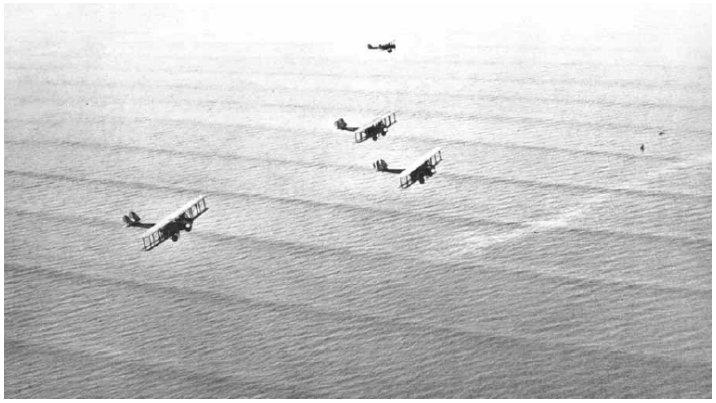

FIGURE I.2 – Photo d'ondes cnoïdales au Panama. Crédits : http ://en.wikipedia.org/wiki/Cnoidal_wave

2 Equations étudiées

Dans cette section nous donnons les différentes équations étudiées ainsi que leurs solutions. Nous avons celles de *KdV* (I.2.1) et de *BBM* (I.2.2) pour le cas en faible profondeur, puis celle de *NLS* (I.2.3) pour le cas en grande profondeur. Ces équations nous serviront de tests sur l'implémentation des différentes méthodes.
Ensuite, nous avons les équations couplées de *Serre* (I.2.4) pour la faible profondeur, ainsi que celles du modèle fortement non linéaire *HOS* (I.2.5) pour la profondeur infinie.
Ces deux derniers modèles seront étudiés plus en détails que les premiers, du fait d'une plus grande complexité de calcul.
A noter, que pour le cas des ondes longues, nous faisons le choix de définir les unités en posant l'accélération de la gravité $g = 1$, ainsi qu'une profondeur d'eau sous le niveau moyen $d = 1$.

2.1 Equation de *Korteweg et de Vries*

Voici le premier modèle de vagues que nous allons étudier.

Equation modèle

Nous devons l'équation dite de *KdV* en premier lieu à Boussinesq [10], dans le cadre de la théorie de l'eau peu profonde. Celle-ci peut s'écrire

$$\frac{\partial \eta}{\partial t} + c_0 \frac{\partial \eta}{\partial x} + \alpha\eta \frac{\partial \eta}{\partial x} + \beta \frac{\partial^3 \eta}{\partial x^3} = 0 \tag{I.1}$$

avec, pour la version linéarisée de l'équation modèle, la relation de dispersion

$$\omega = c_0 k - \beta k^3 \tag{I.2}$$

où $\alpha = \frac{3}{2}\sqrt{\frac{g}{d}}$, $\beta = \frac{d^2 c_0}{6}$ et $c_0 = \sqrt{gd}$ la vitesse de propagation linéaire. Cette équation, qui représente un équilibre entre non linéarité et dispersion, est plus souvent associée à Korteweg et de Vries [48]. Pour information, le terme $c_0 \frac{\partial \eta}{\partial x}$ représente un terme d'advection, tandis que le terme non linéaire $\eta \frac{\partial \eta}{\partial x}$ favorise la formation de fronts raides et le terme $\beta \frac{\partial^3 \eta}{\partial x^3}$ favorise la dispersion.

L'équation de *KdV* possède plusieurs types de solutions dont celles que nous avons citées plus haut, les ondes solitaires et cnoïdales.

Solutions de l'équation : Onde solitaire

La première que nous pouvons citer est l'onde solitaire de la figure I.3, qui se met sous la forme

$$\eta = \frac{H}{\cosh^2\left[\frac{\kappa}{2d}(x - ct)\right]} \tag{I.3}$$

avec $\kappa = \sqrt{\frac{3H}{d}}$, la vitesse de propagation $c = c_0\left(1 + \frac{H}{2d}\right)$ et sa largeur à mi-hauteur de l'ordre de $\frac{1}{\sqrt{H}}$.

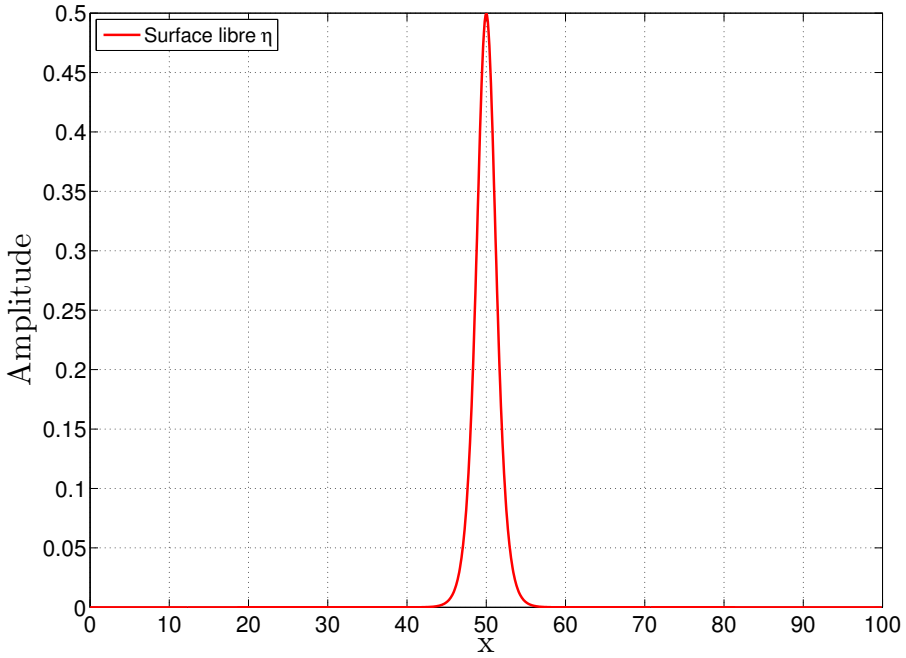

FIGURE I.3 – Onde solitaire d'amplitude $H = 0.5$.

Solutions de l'équation : Onde cnoïdale

Les ondes cnoïdales sont une classe des solutions progressives de l'équation de KdV qui est définie par deux paramètres. Si nous travaillons avec un domaine périodique, il est numériquement préférable d'employer une solution périodique pour tester des équations. De plus, il s'agit de solutions exactes de l'équation considérée, qui peuvent nous servir de tests de propagation. Nous pouvons montrer que cette famille de solutions s'exprime en fonction des paramètres du profil cnoïdal m et de l'amplitude H :

$$\begin{aligned} \eta &= \eta_2 + H\ \mathrm{cn}^2\left[\frac{x - ct}{\Delta} \mid m\right] \\ \eta_2 &= \frac{H}{m}\left(1 - m - \frac{E(m)}{K(m)}\right) \end{aligned} \tag{I.4}$$

avec $\Delta = \dfrac{\lambda}{2K(m)}$. Il faut faire appel aux fonctions Jacobiennes elliptiques $cn(u \mid m)$ (où u est la discrétisation en espace et m le paramètre cnoïdal), ainsi qu'aux intégrales elliptiques complètes de première et seconde espèces $K(m)$ et $E(m)$: ces fonctions sont tabulées dans la littérature.
La vitesse de propagation c, la longueur d'onde λ et la période τ sont décrites par les relations

suivantes

$$\begin{aligned} c &= c_0 \left[1 + \frac{H}{md}\left(1 - \frac{1}{2}m - \frac{3}{2}\frac{E(m)}{K(m)}\right)\right] \\ \lambda &= d\sqrt{\frac{16}{3}\frac{md}{H}}K(m) \\ \tau &= \frac{\lambda}{c} \end{aligned} \tag{I.5}$$

Lorsque $m \to 0$, alors l'onde cnoïdale tend vers une onde sinusoïdale

$$\eta \simeq \frac{H}{2}\cos\left[\frac{2\pi}{\lambda}(x - ct)\right] \tag{I.6}$$

et lorsque nous prenons le cas $m \to 1$, la longueur d'onde $\lambda \to \infty$. De ce fait, l'onde cnoïdale tend vers l'onde solitaire de KdV (creux très plat et crête pentue)

$$\eta = H\ \mathrm{sech}^2\left[\sqrt{\frac{3H}{4d^3}}(x - ct)\right] \tag{I.7}$$

comme nous le voyons sur la figure I.4. Toutes les valeurs intermédiaires de ce paramètre nous permettent de décrire tous les profils souhaités.

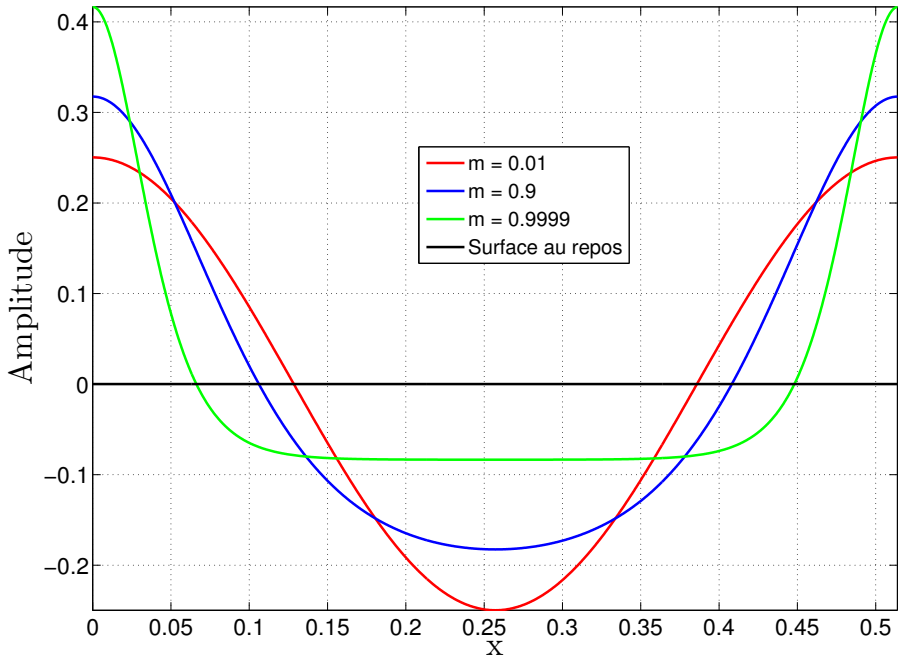

FIGURE I.4 – Différentes ondes cnoïdales à amplitude totale fixée, pour les paramètres $m = 0.01$ (rouge), $m = 0.9$ (bleu) et $m = 0.9999$ (vert).

2.2 Equation de *Benjamin, Bona et Mahony*

Equation modèle

Le terme $d^2 c_0 \dfrac{\partial \eta}{\partial x^3}$ de l'équation de KdV en (I.1) est équivalent au terme $-d^2 \dfrac{\partial \eta}{\partial x^2 \partial t}$, puisque les termes négligés sont du même ordre de grandeur que ceux qui l'ont déjà été pour obtenir

l'équation initiale. Nous obtenons alors l'équation

$$\frac{\partial \eta}{\partial t} + c_0 \frac{\partial \eta}{\partial x} + \frac{3}{2}\sqrt{\frac{g}{d}}\eta \frac{\partial \eta}{\partial x} - \frac{d^2}{6}\frac{\partial^3 \eta}{\partial x^2 \partial t} = 0 \tag{I.8}$$

qui se nomme *BBM* pour *Benjamin, Bona et Mahony* [3] (ou encore *RLW* pour *Regularized Long Wave*) et a, pour la version linéarisée de l'équation modèle, la relation de dispersion

$$\omega = \frac{k c_0}{1 + \frac{k^2 d^2}{6}} \tag{I.9}$$

Cette équation est considérée comme une amélioration de l'équation de *KdV* pour la simulation d'ondes longues et se révèle être aussi plus stable pour les grands nombres d'ondes [25]. Cela s'explique par le fait que pour l'équation de *KdV*, nous avons la relation de dispersion (I.2) qui tend vers l'infini pour $k \to \infty$, alors que pour l'équation de *BBM*, la relation de dispersion en (I.9) tend vers zéro, ce qui fait que nous restons dans une zone *stable*.

Solution de l'équation

Pour cette équation il existe aussi une solution de type cnoïdal :

$$\begin{aligned} \eta &= \eta_2 + H \ \mathrm{cn}^2 \left[\frac{x - ct}{\Delta} \mid m \right] \\ \eta_2 &= \frac{H}{m}\left(1 - m - \frac{E(m)}{K(m)}\right) \end{aligned} \tag{I.10}$$

La différence avec celle qui est solution de l'équation de *KdV* se situe au niveau de l'équation de la longueur d'onde λ (cf. [25] page 715)

$$\begin{aligned} \Delta &= \frac{\lambda}{2K(m)} \\ \lambda &= d\sqrt{\frac{16}{3}\frac{mh}{H}\frac{c}{c_0}}K(m) \\ c &= c_0\left[1 + \frac{H}{md}\left(1 - \frac{1}{2}m - \frac{3}{2}\frac{E(m)}{K(m)}\right)\right] \\ \tau &= \frac{\lambda}{c} \end{aligned} \tag{I.11}$$

Lorsque $m \to 1$, la longueur d'onde $\lambda \to \infty$ et donc l'onde cnoïdale tend vers l'onde solitaire de *BBM*

$$\eta = H \ \mathrm{sech}^2 \left[\sqrt{\frac{3Hc_0}{4d^3 c}} (x - ct) \right] \tag{I.12}$$

dont la différence avec celle de *KdV* est l'ajout du rapport $\frac{c_0}{c}$ entre les vitesses de propagation.

2.3 Equation de *Schrödinger Non Linéaire*

Avec l'équation de *Schrödinger Non Linéaire* nous étudions une équation utilisée dans de nombreux champs de recherche, comme en mécanique quantique avec le condensat de Bose-Einstein [52] ou les fluides et solides quantiques [39], ou encore en optique non linéaire [71]. En mécanique des fluides, cette équation permet d'étudier l'évolution et l'interaction d'un paquet d'ondes, avec une modulation d'amplitude de type instabilité de Benjamin-Feir [43]. Cela peut nous permettre de générer des solutions extrêmes et localisées, comme cela a été obtenu expérimentalement [18].

Equation modèle et solution

L'équation classique de Schrödinger Non Linéaire, notée *NLS*,

$$\psi_t + c_g\psi_x + \frac{\mathrm{i}}{4}c_g k_0^{-1}\psi_{xx} + \frac{\mathrm{i}}{2}\omega_0 k_0^2|\psi|^2\psi = 0 \tag{I.13}$$

et dont la dérivation est disponible dans [24], est décrite pour une enveloppe complexe $\psi(x,t)$ de la surface libre η, dans le cas d'une profondeur infinie, telle que

$$\eta(x,t) = \mathrm{Re}\{\psi(x,t)\mathrm{e}^{\mathrm{i}(k_0x-\omega_0t)}\} \tag{I.14}$$

avec $\omega_0 = \sqrt{gk_0}$ et $c_g = \dfrac{\omega_0}{2k_0}$ la vitesse de groupe linéaire.
Cette équation admet parmi ces solutions, celle de type *soliton enveloppe*, d'amplitude a,

$$\psi = a\ \mathrm{sech}\left(\sqrt{2}ak_0^2(x-c_gt)\right)\mathrm{e}^{-\mathrm{i}a^2k_0^2\frac{\omega_0}{4}t} \tag{I.15}$$

qui est représentée en bleu sur la figure I.5, pour les paramètres $k_0 = 1$ et $a = 0.01$. Dans nos simulations nous prendrons toujours $k_0 = 1$.

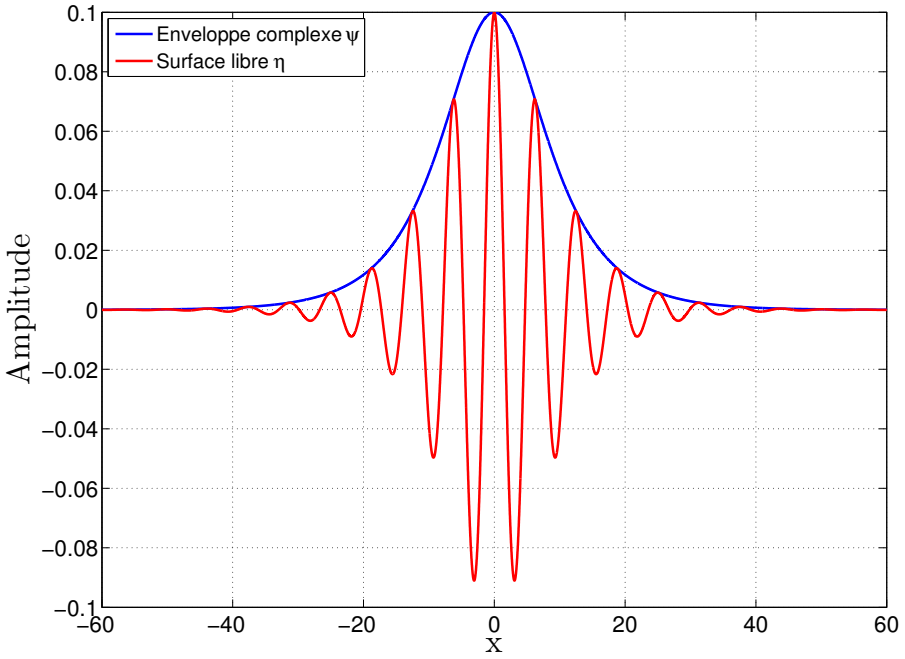

FIGURE I.5 – Profil initial pour les paramètres $a = 0.01$ et $k_0 = 1$. Enveloppe complexe en bleu et surface libre en rouge.

Repère d'étude mobile

En posant le changement de variable $\widetilde{x} = x - c_g t$ et $\widetilde{t} = t$, nous avons

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \widetilde{x}}\frac{\partial \widetilde{x}}{\partial x} + \frac{\partial}{\partial \widetilde{t}}\frac{\partial \widetilde{t}}{\partial x}, \qquad \frac{\partial}{\partial t} = \frac{\partial}{\partial \widetilde{x}}\frac{\partial \widetilde{x}}{\partial t} + \frac{\partial}{\partial \widetilde{t}}\frac{\partial \widetilde{t}}{\partial t} \tag{I.16}$$

et donc

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \widetilde{x}}, \qquad \frac{\partial}{\partial t} = -c_0\frac{\partial}{\partial \widetilde{x}} + \frac{\partial}{\partial \widetilde{t}} \tag{I.17}$$

En laissant tomber la notation en tildes nous obtenons, dans le repère mobile de vitesse c_g, la nouvelle équation de *NLS*

$$\psi_t + \frac{\mathrm{i}}{4}c_g k_0^{-1}\psi_{xx} + \frac{\mathrm{i}}{2}\omega_0 k_0^2|\psi|^2\psi = 0 \tag{I.18}$$

Ce changement de repère nous permet de nous mettre dans le référentiel se déplaçant à la vitesse c_g.

2.4 Equations de *Serre*

Serre [65] a développé une théorie plus générale que celle décrite par les équations précédentes. Elle s'appuie sur un système d'équations moyennées sur la colonne d'eau au-dessus d'un fond graduellement varié. Cette théorie prend en compte des effets convectifs associés à une forte accélération verticale du fluide. Elle a été redécouverte indépendamment par Su et Gardner [70], ou encore par Green, Laws et Naghdi [38]. Elle peut aussi être appelée *weakly-dispersive fully-nonlinear approximation* [82] et nous pouvons citer le travail de Seabra-Santos *et al.* [64] qui ont généralisé ces équations à un fond quelconque.

Equations modèles

Dans notre étude nous considérons un fond constant, donc $h(x,t) = d + \eta(x,t)$. Nous avons le système

$$\begin{aligned} &h_t + [hu]_x = 0 \\ &q_t = \left[\frac{1}{2}u^2 - gh + \frac{1}{2}h^2u_x^2 - uq\right]_x \\ &q = u - \frac{1}{3}h^{-1}[h^3u_x]_x \end{aligned} \tag{I.19}$$

avec les équations sur la hauteur h et sur une quantité q, qui est reliée par la troisième relation à la vitesse horizontale u moyennée sur la colonne d'eau. Le détail des calculs pour obtenir ces équations est disponible dans [64]. Nous pouvons mettre (I.19) sous la forme généralisée [21],

$$\begin{aligned} &h_t + [hu]_x = 0 \\ &u_t + uu_x + gh_x + \frac{1}{3}h^{-1}[h^2\Gamma]_x = 0 \end{aligned} \tag{I.20}$$

et nous retrouvons les équations de Serre classiques avec $\Gamma = h(u_x^2 - uu_{xx} - u_{xt})$ qui est l'accélération verticale à la surface.

Enfin, nous avons aussi, pour la version linéarisée des équations modèles, la relation de dispersion

$$\omega^2 = \frac{gk^2 d}{1 + \frac{k^2 d^2}{3}} \tag{I.21}$$

Nous faisons le choix d'étudier ce modèle pour la *complexité* due au système d'équations couplées et à la relation entre les différentes quantités q, u et u_x.

Solution des équations

Ces équations admettent une solution d'onde progressive cnoïdale $(2\pi/k)$-périodique

$$\begin{aligned} \eta &= a\left(\frac{\mathrm{dn}^2\left(\frac{1}{2}\chi\left(x-ct\right)\mid m\right) - \frac{E(m)}{K(m)}}{1-\frac{E(m)}{K(m)}}\right) = a - H\ \mathrm{sn}^2\left(\frac{1}{2}\chi\left(x-ct\right)\mid m\right) \\ u &= \frac{c\ \eta}{d+\eta} \end{aligned} \tag{I.22}$$

Les paramètres sont reliés entre eux via les relations

$$\begin{aligned} k &= \frac{\pi\chi}{2K(m)} = \frac{2\pi}{L} \\ H &= \frac{maK(m)}{K(m)-E(m)} \\ (\chi d)^2 &= \frac{gH}{\frac{1}{3}mc^2} \\ m &= \frac{gH(d+a)(d+a-H)}{g(d+a)^2(d+a-H) - d^2c^2} \end{aligned} \tag{I.23}$$

où χ représente le nombre d'onde et c est la vitesse de phase observée dans le référentiel de flux nul, c'est-à-dire où $\int_0^L hu\ \mathrm{d}x = 0$ car l'équation sur h (I.20) est une équation de *conservation* . Pour le cas limite $m \to 1$, nous avons $K \to \infty$, $E/K \to 0$, $k \to 0$, $H \to a$ et par conséquent, une onde solitaire est retrouvée pour le cas des équations de *Serre*

$$\begin{aligned} \eta &= a\ \mathrm{sech}^2\left(\frac{\kappa}{2}\left(x-ct\right)\right) \\ c^2 &= g\left(d+a\right) \\ \frac{a}{d} &= \frac{\frac{1}{3}\left(\chi d\right)^2}{1-\frac{1}{3}\left(\chi d\right)^2} \end{aligned} \tag{I.24}$$

Lois de conservations

Lorsque nous simulons des vagues, afin de pouvoir estimer si le résultat obtenu est conforme à ce que nous attendons physiquement, nous ne pouvons pas nous contenter de regarder le profil final de l'onde. Nous devons regarder en plus l'évolution de certaines quantités qui *doivent* être conservées dans le temps.

Définition d'une loi de conservation

Lorsque nous avons une équation modèle, si nous voulons obtenir les relations définissant les quantités conservées, nous l'écrivons comme (I.25) en séparant les dérivées en temps t et en espace x, puis en l'intégrant sur le domaine d'étude L.

$$T_t + X_x = 0 \quad \rightarrow \quad \int_0^L [T_t + X_x]\,\mathrm{d}x = 0 \tag{I.25}$$

Puisque nous avons séparé les deux dépendances, nous pouvons séparer l'intégrale en deux pour obtenir

$$\int_0^L T_t\ \mathrm{d}x + \int_0^L X_x\ \mathrm{d}x = 0 \tag{I.26}$$

Comme nous simulons des vagues dans une boîte de calcul périodique en espace, l'intégrale de la quantité X sur la variable d'espace x est nulle. Nous pouvons donc intervertir l'intégrale sur x et la dérivée sur t, ce qui nous donne

$$\frac{\partial}{\partial t}\int_0^L T\ \mathrm{d}x = 0 \quad \rightarrow \quad \int_0^L T\ \mathrm{d}x = \text{cste} \tag{I.27}$$

Cette dernière relation implique que la quantité T soit conservée. De ce fait, pour avoir un critère permettant de juger de la qualité du résultat, nous surveillerons l'évolution de différentes quantités T tout au long de nos simulations. En effet, puisque les équations étudiées possèdent un certain nombre d'invariants, exprimés sous forme d'intégrales, les solutions de ces équations doivent satisfaire ces propriétés intégrales.

Lois de conservations pour les équations de Serre

Existence de lois de conservations et de constantes du mouvement [15, 57] :

$$\begin{aligned}
\partial_t\left(h\right) + \partial_x\left(hu\right) &= 0 \\
\partial_t\left(hu\right) + \partial_x\left(\frac{1}{2}gh^2 + hu^2 - \frac{1}{3}h^3u_{xt} + \frac{1}{3}h^3u_x^2 - \frac{1}{3}h^3uu_{xx}\right) &= 0 \\
\partial_t\left(\frac{1}{2}hu^2 + \frac{1}{2}gh^2 + \frac{1}{6}h^3u_x^2\right) + \partial_x\left(hu\left(gh + \frac{1}{2}u^2 + \frac{1}{2}h^2u_x^2 - \frac{1}{3}h^2\left(u_{xt} + uu_{xx}\right)\right)\right) &= 0 \\
\partial_t\left(u - hh_xu_x - \frac{1}{3}h^2u_{xx}\right) + \partial_x\left(gh + \frac{1}{2}u^2 - hh_xuu_x - \frac{1}{2}h^2u_x^2 - \frac{1}{3}h^2uu_{xx}\right) &= 0
\end{aligned} \tag{I.28}$$

La première équation correspond à la conservation de la masse, la seconde à la conservation de la quantité de mouvement, la troisième à la conservation de l'énergie et la quatrième correspond à la conservation de la vorticité potentielle.

2.5 Equations du modèle *High-Order Spectral*

Cette théorie des vagues sera la dernière que nous étudierons. C'est une méthode fortement non linéaire dite « High-Order Spectral », notée *HOS*, initialement développée par West *et al.* [80] et Dommermuth et Yue [26].

Equations modèles

Nous avons les équations sur la surface libre η et sur le potentiel à la surface ϕ^S

$$\begin{aligned} \eta_t &= \left(1 + (\nabla_h \eta)^2\right) \cdot W - \nabla_h \phi^S \cdot \nabla_h \eta \\ \phi_t^S &= -g\eta - \frac{1}{2}(\nabla_h \phi^S)^2 + \frac{1}{2}(1 + (\nabla_h \eta)^2) W^2 \end{aligned} \tag{I.29}$$

avec W la vitesse verticale au niveau de la surface libre.
Le détail des étapes menant à ces équations est disponible dans l'annexe A. Ce modèle d'équations de vagues est *Hamiltonien* et possède des invariants, dont l'Hamiltonien qui correspond à l'énergie mécanique totale (énergies cinétique et potentielle).

Solution des équations

Contrairement aux précédents modèles, ici nous n'avons pas de solution cnoïdale. Nous faisons appel à des ondes de *Stokes*. Un paramètre important de ces ondes est la cambrure, qui est le rapport entre hauteur et longueur d'onde. Sa valeur varie en général entre 0 et une valeur limite très proche du déferlement. Dans notre étude nous regardons les cambrures suivantes : 0.10 (faible pente), 0.20 (moyenne pente), 0.30 (grande pente) et 0.40. Ces différentes conditions initiales sont représentées à la figure I.6. Elles sont obtenues à l'aide de méthodes numériques en *quadruple précision* [31].

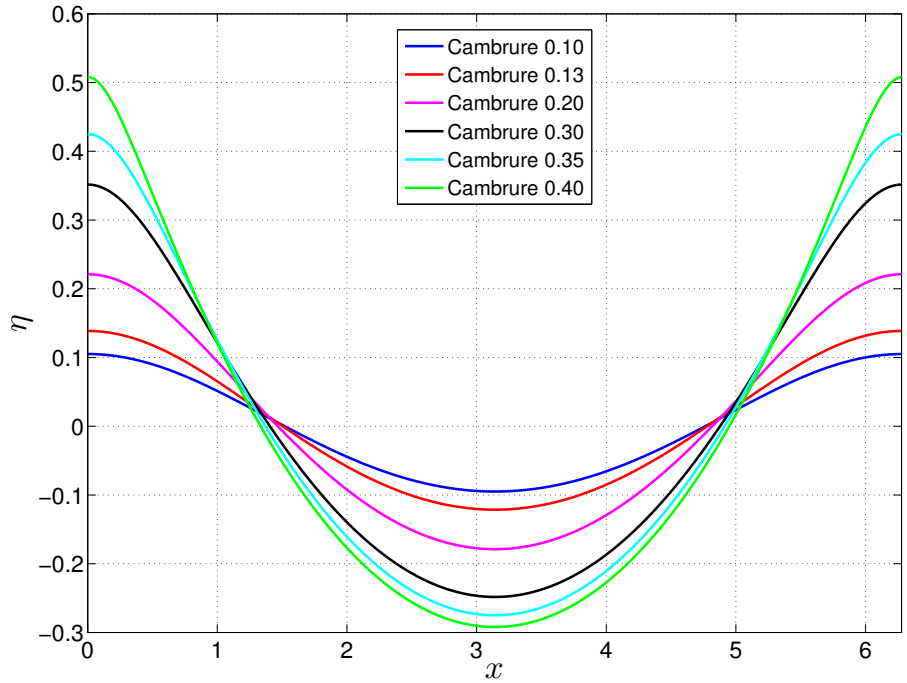

FIGURE I.6 – Différents profils initiaux de la surface libre η pour plusieurs cambrures.

Chapitre II

Méthodes spectrales et temporelles

Dans la première partie de ce chapitre, nous donnons un aperçu des méthodes spectrales en II.1.1 et plus précisément de la discrétisation spatiale en II.1.2. Nous expliquons aussi un point très important pour ces méthodes, la prévention des erreurs numériques en II.1.3. Enfin, nous appliquons un changement d'espace aux équations considérées ici en II.1.4.
Dans la seconde partie du chapitre consacrée à la résolution temporelle, après un rapide état de l'art en II.2.1, nous introduisons en II.2.2 les schémas numériques utilisés dans ce manuscrit, et en II.2.3 la notion de pas de temps variable.

1 Méthodes spectrales

Pour comprendre l'importance que les méthodes spectrales ont de nos jours, il faut savoir qu'une grande partie des simulations de la mécanique des fluides qui sert à la modélisation de prévisions météorologiques est basée sur ces elles. Comme autres utilisations, nous pouvons aussi citer les champs d'études en turbulence et la sismologie [33]. Ces méthodes servent aussi dans la simulation d'équations non linéaires des vagues, comme avec les équations de *Korteweg de Vries* ou de *Schrödinger Non Linéaire* (cette dernière étant aussi fortement utilisée en optique) ou encore pour la MagnétoHydroDynamique. Les avantages principaux de ces méthodes sont que la précision des dérivées de fonctions est accrue si celle de départ est suffisamment régulière et que nous avons un calcul efficace grâce aux algorithmes rapides, dits *FFT*.

1.1 Tour d'horizon

Dans le cadre de notre étude nous souhaitons résoudre des équations non linéaires aux dérivées partielles, ce qui ne peut être effectué que numériquement. Pour cela, il existe quatre grandes familles de méthodes, à savoir : les éléments finis, les volumes finis, les différences finies et les méthodes spectrales. Pour avoir plus de détails, voir [73].
Le nom des méthodes dites *spectrales* vient du mot « spectre », qui fait initialement référence à la décomposition de la lumière en plusieurs longueurs d'ondes par un prisme. Par la suite, ce terme est resté lié au travail de la physique ondulatoire et de la décomposition en fréquences. Cet espace dit spectral ou modal est aussi appelé l'espace de *Fourier*.
La différence majeure entre les trois méthodes dites *finies* et les méthodes spectrales vient du fait que les premières sont dites *locales*, les dérivées en un point étant obtenues grâce aux valeurs des points proches, alors que celles qui nous intéressent, les méthodes *spectrales*, sont dites *globales*, parce que les valeurs prises pas les fonctions en chaque point éloigné du point considéré sont très importantes, du fait d'un changement d'espace que nous réalisons, entre l'espace physique et l'espace spectral.

Méthodes spectrales et pseudo-spectrales

Sans rentrer dans les détails, puisque cela est clairement expliqué dans la référence indiquée au début du chapitre, nous donnons ici un aperçu de la mise en place d'une méthode spectrale. Admettons que nous ayons un certain signal $f(x)$. Une méthode spectrale consiste à utiliser une décomposition de la forme

$$f(x) = \sum_{n=0}^{N} A_n \Psi_n(x) \tag{II.1}$$

où les Ψ_n sont les fonctions de base (par exemple des cosinus) et les A_n sont des coefficients qui peuvent être calculés de différentes manières, dont celle qui nous intéresse, la méthode *pseudo-spectrale* ou encore dite de *collocation* [60]. Son nom vient du fait que les quantités intégrales intervenant dans la résolution sont évaluées en des points fixes du domaine considéré, nommés les points de collocation.

Nous avons une décomposition en série trigonométrique qui nous permet d'employer la Transformation de Fourier Rapide, notée *FFT*, pour passer de l'espace physique à l'espace modal, en un temps de calcul de l'ordre de $N\log(N)$ au lieu d'un coût plus habituel en N^2, où N est le nombre total de points de l'espace pris en compte.

1.2 Discrétisation spatiale

L'espace spectral que nous utilisons est celui de *Fourier*. Voici un petit rappel de certaines définitions et notations que nous utilisons. Pour travailler dans cet espace modal, nous avons :

- Les variables de l'espace physique qui sont notées $\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$, où x et y représentent les axes horizontal et vertical.
- Le vecteur d'onde de l'espace de Fourier qui est noté $\vec{k} = \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}$et de module $k = \left\|\vec{k}\right\|$.
- Une variable dans l'espace physique sera notée sans signe particulier, telle que η, tandis que dans l'espace de Fourier nous la noterons $\hat{\eta}$.

– En général, les termes non linéaires y sont notés $\mathcal{F}\{\cdots\}$. Par exemple, nous notons $\mathcal{F}\{\eta^2\}$ plutôt que $\widehat{\eta^2}$.

Espace Physique

Nous pouvons discrétiser l'espace physique des x continus sur un domaine L et défini par N points

$$\Delta x = \frac{L}{N} \tag{II.2}$$

vers les x discrets x_n, avec un entier n tel que $0 \leq n \leq N-1$,

$$x_n = n\Delta x = \frac{nL}{N} \tag{II.3}$$

Espace Modal

Nous avons la fréquence fondamentale

$$\Delta k = \frac{2\pi}{L} \tag{II.4}$$

et ses harmoniques avec le nombre d'onde

$$k = p\Delta k = p\frac{2\pi}{L} \tag{II.5}$$

avec p un entier tel que $0 \leq p \leq N-1$, où $p = 0$ désigne la fréquence nulle et $p = 1$ désigne le mode fondamental.

Choix de l'échantillonnage

Pour discrétiser notre signal initial, il faut déterminer un nombre de points N suffisamment grand. Soit f_{max} la fréquence maximale du signal, alors la valeur de cette fréquence est donc inférieure à celle d'une certaine fréquence critique, notée k_{Nyquist} définie telle que $k_{\text{Nyquist}} = \dfrac{1}{2\Delta x}$. Il faut donc respecter la relation

$$k_{max} < k_{\text{Nyquist}} \tag{II.6}$$

c'est-à-dire que tous les modes de $\hat{\eta}$ doivent être nuls, donc $\widehat{\eta}(k) = 0$, pour toute fréquence k telle que $|k| \geq k_{\text{Nyquist}}$.

Les p fréquences positives k_+, telle que $0 \leq k_+ \leq k_{\text{Nyquist}}$, sont classées par $1 \leq p \leq \dfrac{N}{2} - 1$ et les p fréquences négatives k_-, telle que $-k_{\text{Nyquist}} \leq k_- \leq 0$, le sont par $\dfrac{N}{2} + 1 \leq p \leq N-1$. Puisque nous discrétisons le signal dans l'espace physique, il est périodique dans l'espace de Fourier. De ce fait, la fonction k (II.5) doit être modifiée selon le signe de la fréquence, en posant

$$k = \begin{cases} p\dfrac{2\pi}{L} & si \quad 0 \leq p \leq \dfrac{N}{2} - 1 \\ 0 & si \quad p = \dfrac{N}{2} = N_{Nyquist} \\ (p-N)\dfrac{2\pi}{L} & si \quad p \geq \dfrac{N}{2} + 1 \end{cases} \tag{II.7}$$

La valeur centrale de k pour $p = \frac{N}{2} = N_{Nyquist}$, que l'on appelle fréquence de *Nyquist*, correspond à la fois à $k = k_{\text{Nyquist}}$ et $k = -k_{\text{Nyquist}}$, puisque c'est la fréquence maximale représentable du signal discret. La valeur de la fonction en cette fréquence étant à la fois positive et négative, nous prenons la valeur moyenne entre ces deux extrêmes, c'est-à-dire zéro. Pour simplifier, nous posons directement $k(N_{Nyquist}) = 0$ dès l'initialisation et non $f(k(N_{Nyquist})) = 0$ lors de chaque calcul. Si nous utilisons la fonction k^2 en posant $k^2(N_{Nyquist}) = 0$ au lieu de $f(k^2(N_{Nyquist})) = 0$, il faut bien faire attention à la « vraie » valeur de $k^2(N_{Nyquist})$ qui est loin d'être nulle, puisque si nous traçons k^2, nous n'avons plus une droite discontinue comme pour k en (II.7), mais une parabole périodique bien continue. Pour les puissances impaires de k nous retrouvons la discontinuité et pour les puissances paires de k nous obtenons des paraboles continues en ce point central.

1.3 Erreurs numériques

Lors de simulations spectrales, nous devons faire attention à l'apparition d'erreurs numériques, appelées *aliasing*, qui peuvent se propager sur tout le résultat et rendre la simulation soit inexacte, soit totalement *instable*.

• Le premier type d'*aliasing* est défini par Fornberg [33] : *"l'erreur d'interpolation (repliement) est donc toujours plus importante que l'erreur de troncature (à moins que les deux ne soient exactement nulles)"*.

• Le second type d'*aliasing* vient du fait qu'une méthode pseudo-spectrale relie des points de collocation, l'espace physique, à des modes dans l'espace de Fourier. Ainsi, une variable définie sur N points de collocation pourra être définie par N modes dans l'espace de Fourier. A chaque fois que nous voulons effectuer un produit de grandeurs exprimées par une transformée de Fourier finie, c'est-à-dire lorsque la somme discrète d'une transformée de Fourier est tronquée à un nombre fini de modes N, nous commettons une erreur. Comme un produit entre deux fonctions discrètes de l'espace physique correspond à une convolution circulaire dans l'espace de Fourier, puisque les spectres y sont périodiques, un produit de deux sommes définies pour les points $0, \cdots, N-1$ donne une somme définie sur $0, \cdots, 2N-1$. Or, si le produit est défini par N composantes, une erreur est obligatoirement créée. Les termes qui devraient être décrits par les composantes $N, \cdots, 2N-1$ le seront de manière erronée par les composantes $0, \cdots, N-1$. Nous parlons alors de *repliement de spectre*, car la composante N aura une influence sur la composante $N-1$, celle en $N+1$ sur celle $N-2$, etc. jusqu'à la composante $2N-1$ qui aura une influence sur la composante 0. C'est une difficulté majeure des méthodes spectrales qui, heureusement, peut être évitée si nous utilisons les outils adéquats.
Ce problème majeur a été contourné en utilisant un lissage. Par exemple, Dommermuth et Yue comme Craig et Sulem ont utilisé une moyenne mobile en cinq points qui est équivalente à un filtre passe bas dans l'espace de Fourier. Cette technique amortit les hautes longueurs d'onde, mais malheureusement aussi les plus basses. Cela n'est donc pas idéal pour des simulations sur des temps longs, puisque l'énergie du système décroîtra avec le temps, jusqu'à parfois disparaître, ce qui n'est pas physiquement acceptable.

Pour se prémunir de ces erreurs, nous employons une autre méthode, plus efficace et sans effet non désiré, le *zero-padding*, qui est donc une méthode d'*anti-aliasing*.

Méthode d'anti-aliasing choisie

La méthode choisie pour éviter ces problèmes d'aliasing est celle du *zero-padding*. C'est une technique d'extension et de remplissage du spectre par zéro (voir [14], chapitre 3) de manière à n'avoir aucune haute fréquence qui puisse se replier sur une plus basse fréquence.

Explications pour l'utilisation de l'anti-aliasing

Admettons que nous ayons un produit de deux variables $\hat{f}$ et $\hat{g}$ à faire dans l'espace modal :
• Nous opérons une transformée de Fourier sur f et g nous donnant les N amplitudes modales $\hat{f}$ et $\hat{g}$;
• Nous augmentons le nombre de modes jusqu'à $N_{\text{zero}} \geq N$. Pour ce faire, les amplitudes modales comprises entre $N-1$ et $N_{\text{zero}}-1$ sont prises égales à zéro. D'où la notion de remplissage du spectre par zéro ;
• Nous effectuons une transformée de Fourier inverse pour revenir sur les grandeurs dans l'espace physique f' et g' définies cette fois sur N_{zero} points de collocation ;
• Nous effectuons le produit $f' \times g'$;
• Nous réalisons une transformée de Fourier donnant les N_{zero} amplitudes modales.
• Enfin, nous ne gardons que les N premières amplitudes qui correspondront aux amplitudes du produit traité contre le repliement.

Afin d'éviter de devoir modifier la taille des vecteurs ou matrices contenant f et g à chaque calcul en utilisant des tableaux dynamiques, ou d'utiliser deux fois plus d'espace en terme de mémoire d'ordinateur avec deux tailles différentes N et N_{zero}, nous modifions légèrement la procédure précédente. Dès le début de la simulation nous prenons un signal avec N_{zero} points, mais dont les valeurs allant de N à $N_{\text{zero}}-1$, qui sont nulles, sont ajoutées artificiellement par cette méthode. Enfin, lors de chaque calcul de non linéarité, ces points supplémentaires seront *toujours* mis à zéro.

Choix de l'extension pour l'anti-aliasing

Il nous faut désormais choisir le nombre de points N_{zero} afin d'éliminer la possibilité d'avoir des termes repliés. Pour cela, nous faisons appel à la règle communément admise des $(M+1)$ *moitiés* pour un produit d'ordre M. Pour un signal contenant N points, le nouveau nombre de points à prendre en compte sera défini par

$$N_{\text{zero}} = \frac{(M+1)}{2} N \tag{II.8}$$

Ainsi, le spectre doit être étendu par au moins un facteur trois-demi pour les non linéarités quadratiques, un facteur quatre-demi pour les non linéarités cubiques, etc.

Exemple de traitement contre l'aliasing

Pour bien comprendre cette partie importante pour les méthodes spectrales, considérons une non linéarité quadratique ($M=2$). Si nous avons un signal avec $N=128$ points de fréquences *utiles* (nulle, positives et négatives) dans le spectre de Fourier, alors il nous faudra au minimum $N_{\text{zero}} = \dfrac{2+1}{2} N = 192$ points. Comme nous utilisons des puissances de 2 comme nombre de

points pour des raisons numériques, dans ce cas nous prendrions 256 points.
Au passage, il faut noter que ce n'est pas entièrement équivalent de calculer f^4 avec la règle des cinq-demis et de calculer $f^2 \times f^2$ en appliquant deux fois la règle des trois-demi.
Enfin, comme le spectre de Fourier des fonctions régulières décroît exponentiellement, l'erreur introduite par les applications multiples de la règle des trois-demi sur $(f^2)^2$ est comparable à l'erreur d'arrondi, si la discrétisation de f est suffisamment raffinée [22, 36].

Utilité de ce traitement d'anti-aliasing

Sur la figure II.1 nous représentons le spectre de Fourier de la surface libre η après quelques périodes de simulations pour le modèle de *BBM* et le profil cnoïdal de paramètres $m = 0.99$ et $H = 0.5$. Sur la partie gauche du graphique nous avons les fréquences positives et sur la partie droite les fréquences négatives. La courbe bleue représente le résultat d'une simulation réalisée avec le traitement contre l'aliasing. Nous voyons bien que les hautes fréquences sont *noyées* dans l'erreur machine de l'ordre de 10^{-16}.
La courbe rouge est obtenue pour la même simulation sans le traitement préventif. Nous observons que dans ce cas, les hautes fréquences qui devraient être nulles ne le sont plus. Nous avons des erreurs numériques importantes, de l'ordre de 10^{-12}, ce qui n'est pas physiquement acceptable.

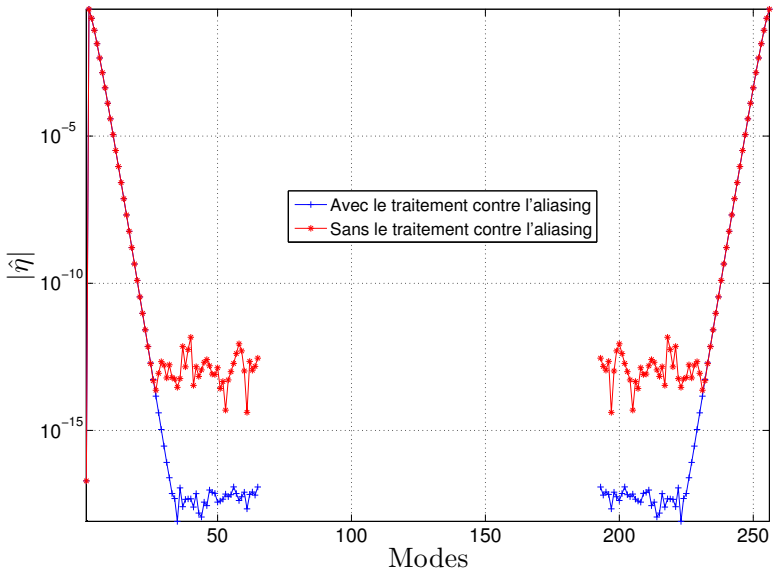

FIGURE II.1 – Spectre de Fourier de la surface libre après quelques périodes de simulations pour le modèle de *BBM*. Le cas avec l'anti-aliasing est en bleu et le cas sans anti-aliasing est en rouge.

Les modes centraux, 65 à 193, pour lesquels nous voyons qu'il n'y a pas de valeurs sont ceux qui sont ajoutés et mis à zéro par la méthode d'anti-aliasing.

1.4 Reformulations des équations des vagues dans l'espace de Fourier

1.4.a *Equation de Korteweg et de Vries*

Dans l'espace modal de Fourier, l'équation (I.1) se met sous la forme

$$\widehat{\eta}_t + \mathrm{i}\left(c_0 k - k^3 \frac{d^2 c_0}{6}\right)\widehat{\eta} + \mathrm{i}k\frac{3}{4}\sqrt{\frac{g}{d}}\mathcal{F}\left\{\eta^2\right\} = 0 \tag{II.9}$$

où $\hat{\eta}$ représente la variable η de l'espace physique dans l'espace de Fourier.

1.4.b *Equation de Benjamin, Bona et Mahony*

Dans l'espace de Fourier, l'équation de *BBM* (I.8) se met sous la forme

$$\widehat{\eta}_t + c_0 \mathrm{i}k\widehat{\eta} + \frac{3}{4}\sqrt{\frac{g}{d}}\mathrm{i}k\mathcal{F}\left\{\eta^2\right\} - \frac{d^2}{6}(\mathrm{i}k)^2\widehat{\eta}_t = 0 \tag{II.10}$$

1.4.c *Equation de Schrödinger Non Linéaire*

Pour un repère d'étude fixe, l'équation de *NLS* (I.13) se met sous la forme

$$\widehat{\psi}_t + c_g \mathrm{i}k\widehat{\psi} + \frac{\mathrm{i}}{4}c_g k_0^{-1}(\mathrm{i}k)^2\widehat{\psi} + \frac{\mathrm{i}}{2}\omega_0 k_0^2 \mathcal{F}\left\{|\psi|^2\psi\right\} = 0 \tag{II.11}$$

tandis que pour un repère d'étude mobile, l'équation (I.18) devient

$$\widehat{\psi}_t + \frac{\mathrm{i}}{4}c_g k_0^{-1}(\mathrm{i}k)^2\widehat{\psi} + \frac{\mathrm{i}}{2}\omega_0 k_0^2 \mathcal{F}\left\{|\psi|^2\psi\right\} = 0 \tag{II.12}$$

1.4.d *Equations de Serre*

Nous pouvons reécrire le système (I.19) dans l'espace de Fourier

$$\begin{aligned}
\widehat{h}_t &= -\mathrm{i}k\ \mathcal{F}\{hu\} \\
\widehat{q}_t &= \mathrm{i}k\ \mathcal{F}\left\{\frac{1}{2}u^2 - gh + \frac{1}{2}h^2u_x^2 - uq\right\} \\
\widehat{q} &= \widehat{u} - \mathcal{F}\left\{\frac{1}{3}h^{-1}[h^3u_x]_x\right\}
\end{aligned} \tag{II.13}$$

avec le développement $-\frac{1}{3}h^{-1}[h^3u_x]_x = -hh_xu_x - \frac{1}{3}h^2u_{xx}$.

Dans les paragraphes suivants, nous donnons quelques éléments de compréhension numérique.

Préférence entre η et h

D'un point de vue numérique, il est préférable de travailler avec la variable η qui a une moyenne nulle, plutôt qu'avec h qui a une moyenne différente de zéro. De ce fait, le système (I.19) dans l'espace physique devient

$$\begin{aligned} \eta_t &= -\left[(\eta+d)\,u\right]_x = -du_x - (\eta u)_x \\ q_t &= \left[\frac{1}{2}u^2 - g\,(\eta+d) + \frac{1}{2}\,(\eta+d)^2\,u_x^2 - uq\right]_x \\ q &= u - \frac{1}{3}\,(\eta+d)^{-1}\,[h^3 u_x]_x \end{aligned} \tag{II.14}$$

et dans l'espace de Fourier, le système (II.13) devient

$$\begin{aligned} \widehat{\eta}_t &= -\mathrm{i}kd\;\widehat{u} - \mathrm{i}k\;\mathcal{F}\left\{\eta u\right\} \\ \widehat{q}_t &= \mathrm{i}k\;\mathcal{F}\left\{\frac{1}{2}u^2 - gh + \frac{1}{2}h^2u_x^2 - uq\right\} \\ \widehat{q} &= \widehat{u} - \mathcal{F}\left\{\frac{1}{3}h^{-1}[h^3u_x]_x\right\} \end{aligned} \tag{II.15}$$

Première approche pour le calcul de u

En développant l'équation sur q en (II.14), nous obtenons l'équation différentielle pour u du second ordre à coefficients variables

$$q = u - hh_xu_x - \frac{1}{3}h^2u_{xx} = F\left(u, u_x, u_{xx}, h, h_x\right) \tag{II.16}$$

Nous ne pouvons que la résoudre numériquement, en la mettant sous la forme

$$u = q + hh_xu_x + \frac{1}{3}h^2u_{xx} = \tilde{F}\left(u, u_x, u_{xx}, h, h_x\right) \tag{II.17}$$

puis en itérant de manière récursive, par la méthode du point fixe, en utilisant la formulation (II.18) et le critère (II.19) sur la précision ε désirée.

$$u^{(n+1)} = q + hh_xu_x^{(n)} + \frac{1}{3}h^2u_{xx}^{(n)} \tag{II.18}$$

$$\left\|u^{(n+1)} - u^{(n)}|\right\| < \varepsilon \tag{II.19}$$

Malheureusement, ainsi, le code numérique s'avère *instable*. En effet, à gauche et à droite de l'équation (II.18) nous n'avons pas le même ordre en u. A gauche nous sommes en u et à droite en u_{xx}, ce qui fait que pour calculer $u^{(n+1)}$ il faut dériver deux fois $u^{(n)}$, c'est-à-dire faire le produit $k^2\hat{u}$ dans l'espace de Fourier. Donc les erreurs numériques sur u vont être fortement amplifiées par le pré-facteur k^2, ce qui va faire *écrouler* la résolution. Pour y remédier, il faut que des deux côtés de l'équation nous ayons au moins le même ordre en u pour ne pas avoir à faire de calculs supplémentaires.

Seconde approche pour le calcul de u

Nous avons $h(x,t) = d + \eta(x,t)$, donc (II.16) peut se reécrire

$$q = u - (d+\eta)\eta_x u_x - \frac{1}{3}(d+\eta)^2 u_{xx} \tag{II.20}$$

Si η est constant, nous aurions alors quelque chose de la forme

$$q = u - \frac{1}{3}d^2 u_{xx} \tag{II.21}$$

En numérique, une astuce consiste à ajouter zéro, par exemple en ajoutant puis retranchant le terme $\frac{1}{3}d^2u_{xx}$ comme dans

$$\begin{aligned} q &= u - \frac{1}{3}d^2u_{xx} + \left[\frac{1}{3}d^2u_{xx} - (\eta+d)\,\eta_x u_x - \frac{1}{3}(\eta+d)^2\,u_{xx}\right] && \text{(II.22)}\\ &= u - \frac{1}{3}d^2u_{xx} - (\eta+d)\,\eta_x u_x - \frac{1}{3}(\eta+2d)\,\eta u_{xx} && \text{(II.23)} \end{aligned}$$

Cela revient à développer (II.20) avec $h = \eta + d$. En redisposant l'ordre des termes, nous voyons apparaître une équation différentielle du second ordre à coefficients constants

$$u - \frac{1}{3}d^2u_{xx} - q = (\eta+d)\,\eta_x u_x + \frac{1}{3}(\eta+2d)\,\eta u_{xx} \tag{II.24}$$

Dans l'espace de Fourier, cette équation devient

$$\widehat{u} + \frac{k^2}{3}d^2\widehat{u} - \widehat{q} = \mathcal{F}\{h\eta_x u_x + \frac{1}{3}(\eta+2d)\,\eta u_{xx}\} \tag{II.25}$$

ce qui nous permet d'écrire la relation cherchée pour u sous la forme

$$\widehat{u} = \frac{\widehat{q}}{1+\frac{k^2d^2}{3}} + \frac{\mathcal{F}\{h\eta_x u_x + \frac{1}{3}(\eta+2d)\,\eta u_{xx}\}}{1+\frac{k^2d^2}{3}} \tag{II.26}$$

Par itérations successives, nous trouvons la valeur de u, grâce à la relation

$$\widehat{u}^{(n+1)} = \frac{\widehat{q} + \mathcal{F}\left\{h\eta_x u_x + \frac{1}{3}(\eta+2d)\,\eta u_{xx}\right\}^{(n)}}{1+\frac{k^2d^2}{3}} \tag{II.27}$$

et au critère sur la précision ε désirée

$$\left\|u^{(n+1)} - u^{(n)}|\right\| < \varepsilon \tag{II.28}$$

Nous pouvons donc déterminer une approximation de u très fiable. La différence entre cette formulation (II.27) et la précédente (II.18), est que désormais, comme nous divisons par k^2, il y a une atténuation par les hautes fréquences. De ce fait, nous avons un schéma plus *stable*. Nous pouvons donc maintenant passer de la variable q à la variable u à tout moment.

Méthodologie pour passer de la variable q à la variable u entre deux pas de temps

Nous partons du temps $t = t_n$ avec les valeurs connues de $h(t_n)$, $q(t_n)$ et $u(t_n)$. Le système évolue temporellement, grâce à un algorithme d'avancement temporel, jusqu'au sous-pas de temps de calcul t_{n+1} afin d'obtenir les valeurs $h(t_{n+1})$ et $q(t_{n+1})$.

Méthode du point fixe :

1. Nous calculons la valeur de $u(t_{n+1})$ par récurrence, d'après (II.27), en nous donnant comme condition initiale pour u, la dernière valeur connue, soit $u^{(n)} = u(t_n)$.
2. Pour calculer les dérivées $u_x^{(n)}$ et $u_{xx}^{(n)}$, nous passons $u^{(n)}$ dans l'espace de Fourier, multiplions par ik ou $-k^2$, puis revenons dans l'espace physique.
3. Nous gardons fixes les valeurs de $h = h(t_{n+1})$ et $q = q(t_{n+1})$ dans (II.27).
4. Nous utilisons le critère (II.28) pour voir si nous avons déjà obtenu la précision ε désirée.
   - Si oui, la récurrence s'arrête et alors $u = u^{(n+1)}$ et nous pouvons continuer l'évolution temporelle.
   - Si non, nous n'avons pas atteint le point fixe et nous continuons la récurrence en incrémentant n pour calculer $u^{(n+2)}$ avec les valeurs précédentes $u^{(n+1)}$, $u_x^{(n+1)}$ et $u_{xx}^{(n+1)}$. Puis nous recommençons cette dernière étape tant que la précision voulue n'est pas atteinte.

Equations sans dimensions

Multiplions la première équation de (II.15) par ik et la seconde par $\dfrac{\mathrm{i}\omega}{g}$. Cela nous donne les nouvelles équations sans dimensions

$$\begin{aligned} \mathrm{i}k\widehat{\eta}_t - \frac{\omega^2}{g}\widehat{q} &= k^2\ \mathcal{F}\left\{\eta u\right\} + \frac{k^2 d}{1+\frac{k^2d^2}{3}}\mathcal{F}\{h\eta_x u_x + \frac{1}{3}\left(\eta + 2d\right)\eta u_{xx}\} \\ \frac{\mathrm{i}\omega}{g}\widehat{q}_t - \omega k\widehat{\eta} &= -\frac{\omega k}{g}\ \mathcal{F}\left\{\frac{1}{2}u^2 - gh + \frac{1}{2}h^2u_x^2 - uq\right\} \end{aligned} \tag{II.29}$$

1.4.e *Equations du modèle High-Order Spectral*

Afin de séparer parties linéaire et non linéaire du système d'équations du modèle *HOS*, nous pouvons écrire la vitesse verticale sous la forme

$$W = W^{(0)} + \left(W - W^{(0)}\right) \tag{II.30}$$

Ainsi, les conditions de surface libre se reécrivent comme

$$\begin{aligned} \eta_t - W^{(0)} &= B_1 \\ \phi_t^s + g\eta &= B_2 \end{aligned} \tag{II.31}$$

avec les deux membres de droite B_1 et B_2 qui sont des termes purement non linéaires :

$$\begin{aligned} B_1 &= -\nabla\phi_s \cdot \nabla\eta + \left(W - W^{(0)}\right) + \left(1 + |\nabla\eta|^2\right)W \\ B_2 &= -\frac{1}{2}|\nabla\phi^s|^2 + \frac{1}{2}\left(1 + |\nabla\eta|^2\right)W^2 \end{aligned} \tag{II.32}$$

Dans notre étude, les membres B_1 et B_2 contenant les termes non linéaires peuvent être de deux types, selon que nous prenons en compte les équations de West et Watson, ou celles de Dommermuth et Yue pour le calcul des termes en $\left(1+(\nabla_h\eta)^2\right)$, comme expliqué à la fin de l'annexe A.
Notons que nous avons aussi

$$\mathcal{F}\left[W^{(0)}\right] = \mathcal{F}D[\phi^s] = k\,\tanh(kh)\,\mathcal{F}\left[\phi^s\right] \tag{II.33}$$

avec $D \equiv k\,\tanh(kh)$, ce qui fait que nous pouvons écrire les équations d'évolution dans l'espace de Fourier sous la forme

$$\begin{aligned} \frac{\partial}{\partial t}\hat{\eta} - k\,\tanh(kh)\,\hat{\phi^s} &= \hat{B}_1 \\ \frac{\partial}{\partial t}\hat{\phi^s} + g\hat{\eta} &= \hat{B}_2 \end{aligned} \tag{II.34}$$

En faisant appel à la relation de dispersion $\omega^2 = gk\tanh(kh)$, nous pouvons reécrire les équations d'évolutions comme

$$\begin{aligned} \frac{\partial}{\partial t}\hat{\eta} - \omega\left(\frac{\omega}{g}\hat{\phi^s}\right) &= \hat{B}_1 \\ \frac{\partial}{\partial t}\left(\frac{\omega}{g}\hat{\phi^s}\right) + \omega\hat{\eta} &= \frac{\omega}{g}\hat{B}_2 \end{aligned} \tag{II.35}$$

Solution des équations

Pour les équations complètes de ce modèle *HOS*, c'est-à-dire sans approximations, il n'existe pas de solution analytique. Heureusement, nous disposons de solutions représentant des ondes progressives solutions du problème, calculées de manière très précises. Ces solutions sont caractérisées par leur cambrure $\epsilon = ak$, où $a = \dfrac{H}{2}$ et $k = \dfrac{2\pi}{\lambda}$, comme montré sur la figure I.6 en I.2.5. Aux fortes cambrures, les méthodes utilisées montrent quelques limites à prédire correctement l'amplitude spectrale des harmoniques d'ordre élevé, comme nous le voyons sur la figure II.2 représentant les profils initiaux dans l'espace de Fourier. Afin d'éviter la contamination des solutions numériques par ces erreurs initiales, nous veillerons, lors de nos simulations, à filtrer convenablement ces solutions, qui seront les conditions initiales du modèle *HOS*. Nous fixons le nombre maximum de fréquences à garder à l'aide de la fréquence de coupure, noté k_m, dans le tableau II.1.

TABLE II.1 – Fréquence de coupure par cambrure.

ak	k_m	ak	k_m	ak	k_m	ak	k_m	ak	k_m	ak	k_m
0.10	32	0.13	32	0.20	32	0.30	50	0.35	48	0.40	64

Il faut savoir qu'avec le modèle *HOS* utilisé nous ne pouvons pas propager correctement des ondes dont la cambrure dépasse la valeur de 0.30 sur un grand nombre de périodes. Par contre, sur des temps courts cela reste tout à fait possible.

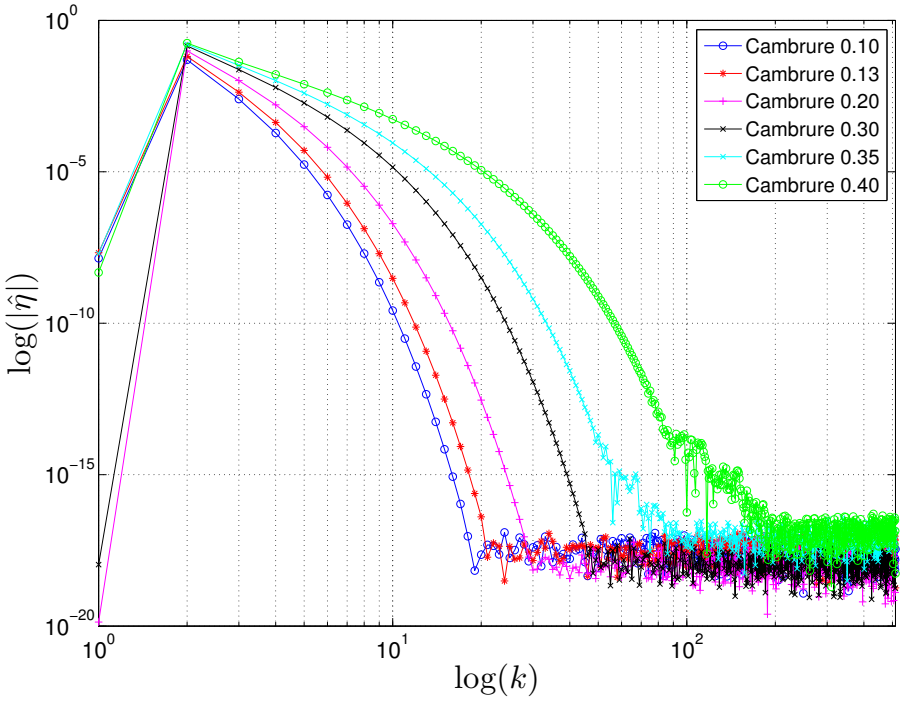

FIGURE II.2 – Différents profils initiaux de la surface libre η pour plusieurs cambrures. Représentations dans l'espace physique à gauche et dans l'espace modal de Fourier à droite.

Ces différentes conditions initiales sont obtenues à l'aide de méthodes numériques en *quadruple précision* [31].

Algorithme temporel de résolution numérique

Au début d'un pas de temps de calcul t_n, nous avons les vecteurs élévation η_n et potentiel de surface ϕ_n dans l'espace physique. En ce qui concerne leurs gradients horizontaux, nous les évaluons dans l'espace de Fourier. La vitesse verticale W est déterminée grâce à un double schéma itératif (équations A.9 et A.12). Une fois que nous avons toutes ces quantités, il ne nous reste plus qu'à faire une avance temporelle classique, comme celles que nous verrons dans la deuxième partie de ce chapitre.

1.4.f *Calcul des termes non linéaires : Essence de l'approche pseudo-spectrale*

Admettons que nous ayons à calculer un terme non linéaire du type $\mathcal{F}\left\{\eta^2\right\}$. Ce calcul ne peut se faire correctement sans changer d'espace de travail. Pour cela, nous passons de la variable spectrale $\hat{\eta}$ à la variable η de l'espace physique. Nous calculons la non linéarité η^2, puis nous retournons dans l'espace de Fourier pour obtenir ce que nous recherchons, à savoir $\mathcal{F}\left\{\eta^2\right\}$. Nous procéderons toujours ainsi pour toutes les non linéarités. Prenons l'exemple des équations modèles de *Serre* vues en II.1.4.d. Pour calculer les termes non linéaires, nous partons des valeurs de l'espace physique, nous faisons les multiplications entre variables, puis revenons dans l'espace de Fourier. Les détails de ces étapes sont montrés ci-dessous, où le terme *FFT* désigne le passage de l'espace physique vers l'espace de Fourier et le terme *iFFT* désigne le passage

inverse.

$$
\begin{aligned}
&hu \xrightarrow[FFT]{} \mathcal{F}\{hu\} \rightarrow \mathrm{i}k\, \mathcal{F}\{hu\} \\
&u \rightarrow u^2 \xrightarrow[FFT]{} \mathcal{F}\left\{u^2\right\} \\
&u \xrightarrow[FFT]{} \widehat{u} \rightarrow \mathrm{i}k\, \widehat{u} \xrightarrow[iFFT]{} u_x \rightarrow u_x^2 \rightarrow h^2 u_x^2 \xrightarrow[FFT]{} \mathcal{F}\left\{h^2 u_x^2\right\} \\
&u \xrightarrow[FFT]{} \widehat{u} \rightarrow \mathrm{i}k\, \widehat{u} \xrightarrow[iFFT]{} u_x \rightarrow h^3 u_x \xrightarrow[FFT]{} \mathcal{F}\left\{h^3 u_x\right\} \rightarrow \mathrm{i}k\, \mathcal{F}\left\{h^3 u_x\right\} \\
&\mathrm{i}k\, \mathcal{F}\left\{h^3 u_x\right\} \xrightarrow[iFFT]{} [h^3 u_x]_x \rightarrow u h^{-1} [h^3 u_x]_x \xrightarrow[FFT]{} \mathcal{F}\left\{u h^{-1} [h^3 u_x]_x\right\}
\end{aligned}
\tag{II.36}
$$

2 Résolution temporelle

Nous souhaitons simuler dans le temps les équations de vagues précédentes. Pour cela, nous avons besoin d'utiliser un intégrateur temporel (explications en II.2.1). Nous allons nous servir de plusieurs méthodes de *Runge-Kutta*. Notre choix s'est orienté vers les modèles développés par Bogacki et Shampine, Dormand et Prince et Verner (décrits en II.2.2) puisqu'ils nous permettront d'utiliser un pas adaptatif efficace (voir II.2.3)

2.1 Tour d'horizon

La plupart des équations différentielles intéressantes sont non linéaires et à quelques exceptions près, elles ne peuvent pas être résolues exactement. Nous devons donc approcher les solutions à l'aide d'approximations numériques. Il existe deux types de méthodes, celles dites *explicites* pour lesquelles nous utilisons les données du système au temps actuel pour trouver la solution à un temps plus grand, et celles dites *implicites*, pour lesquelles nous utilisons les données du système aux temps actuel et suivant pour trouver la solution. Cette dernière approche permet d'être plus stable pour des problèmes assez difficiles à simuler, mais a un coût de calcul nettement supérieur. C'est une des raisons qui fait que nous allons nous concentrer uniquement sur des méthodes explicites.

Nous partons d'une équation de la forme

$$
y'(t) = f(t, y) \qquad \rightarrow \qquad y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y)\, \mathrm{d}t \tag{II.37}
$$

Dans notre notation, $t_{initial}$ et t_{final} sont les temps de départ et de fin de la simulation en entier, t_n et t_{n+1} sont les temps de départ et de fin d'une boucle de temps réalisée avec un pas de temps de calcul noté Δt, tel que $t_{n+1} = t_n + \Delta t$. Nous introduisons aussi les notations ci-dessous :

$$
\begin{aligned}
y_n &= y\left(t_n\right) \\
y_{n+1} &= y\left(t_{n+1}\right) \\
\frac{\mathrm{d}y_n}{\mathrm{d}t} &= y'(t) = f\left(t_n, y_n\right) = f\left(y_n\right) \\
\frac{\mathrm{d}^2 y_{n+1}}{\mathrm{d}t^2} &= y''_{n+1}(t) = f'\left(t_{n+1}, y_{n+1}\right) = f^{(1)}\left(t_{n+1}, y_{n+1}\right)
\end{aligned}
\tag{II.38}
$$

Méthode d'Euler explicite

La méthode d'Euler explicite est une méthode du premier ordre, la plus élémentaire des méthodes explicites pour réaliser une intégration numérique d'équations aux dérivées partielles (*EDP*) et aussi la plus simple des méthodes dites de *Runge-Kutta*. Pour passer du temps t_n au temps t_{n+1} nous avons la relation

$$y_{n+1} = y_n + \Delta t \; f(t_n, y_n) \tag{II.39}$$

Il existe aussi une méthode d'Euler implicite, ce qui veut dire que le terme y_{n+1} se retrouve des deux côtés de l'équation précédente.

Méthodes de Runge-Kutta

Pour obtenir des méthodes d'ordres supérieurs (et plus précises), il est nécessaire d'utiliser plus d'évaluations de fonctions, comme avec les méthodes de *Runge-Kutta*. Un intégrateur temporel de type Runge-Kutta, à l'ordre p, est une méthode numérique qui nous donne une solution exacte du problème pour un polynôme d'ordre inférieur ou égal à p. En analyse numérique, les méthodes de Runge-Kutta représentent une famille importante des méthodes explicites et implicites pour l'approximation de solution d'*EDP*. Ces techniques ont été développées en 1901 par Carl Runge et Martin Wilhelm Kutta.

Méthodes de Runge-Kutta explicites

Les méthodes explicites de Runge-Kutta se notent

$$y_{n+1} = y_n + \Delta t \sum_{i=1}^{s} b_i k_i \tag{II.40}$$

avec les calculs des *dérivées* intermédiaires

$$\begin{aligned}
k_1 &= f\left(t_n, y_n\right) \\
k_2 &= f\left(t_n + c_2 \Delta t, y_n + \Delta t a_{21} k_1\right) \\
k_3 &= f\left(t_n + c_3 \Delta t, y_n + \Delta t \left(a_{31} k_1 + a_{32} k_2\right)\right) \\
&\cdots \\
k_s &= f\left(t_n + c_s \Delta t, y_n + \Delta t \left(a_{s1} k_1 + \cdots + a_{s,s-1} k_{s-1}\right)\right)
\end{aligned} \tag{II.41}$$

où les termes a_{ij}, b_j et c_i sont des constantes déterminées par des relations connues (voir [41] page 173) et s est le nombre d'étages de calculs nécessaires pour trouver la solution. Les coefficients a_{ij}, b_j et c_i sont notés de la manière suivante dans le *tableau de Butcher* (qui porte le nom de John Charles Butcher)

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
-- & -- & -- & -- & - - - & -- \\
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array} \tag{II.42}$$

Exemple : Méthode de Runge-Kutta d'ordre 4

Une des méthodes les plus courantes est celle que l'on nomme *RK4*. C'est en fait une méthode à l'ordre 4, ce qui veut dire que l'erreur par pas de calcul est d'ordre Δt^5 puisque satisfait la condition suivante, avec $p = 4$,

$$\|y_{n+1} - y(t_n + \Delta t)\| \leq \mathcal{O}\left(\Delta t^{p+1}\right) \tag{II.43}$$

Au final, la solution de l'équation d'évolution s'écrit

$$y_{n+1} = y_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{II.44}$$

avec les approximations intermédiaires des différentes dérivées k_i, pour $1 \leq i \leq 4$,

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\left(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}k_1\right) \\ k_3 &= f\left(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}k_2\right) \\ k_4 &= f(t_n + \Delta t, y_n + k_3) \end{aligned} \tag{II.45}$$

Méthodes de Runge-Kutta adaptatives

Les méthodes dites *adaptatives* permettent d'estimer l'erreur locale sur un pas de temps de calcul. Cela est possible en utilisant deux méthodes en une, la première d'ordre p et la seconde d'ordre, au plus, $p - 1$. La solution d'ordre $p - 1$, notée $\tilde{y}_{n+1}$, est donnée par

$$\tilde{y}_{n+1} = y_n + \Delta t \sum_{i=1}^{s} \tilde{b}_i k_i \tag{II.46}$$

où les k_i sont les mêmes que ceux calculés en (II.40) pour l'ordre supérieur p et où il n'y a que les pré-facteurs $\tilde{b}_i$ qui changent. Cette solution d'ordre inférieur nous permet d'estimer l'erreur locale

$$e_{n+1} = y_{n+1} - \tilde{y}_{n+1} = \Delta t \sum_{i=1}^{s} (b_i - \tilde{b}_i) k_i = -\Delta t \sum_{i=1}^{s} \tilde{d}_i k_i \tag{II.47}$$

à l'aide du vecteur $\tilde{d}_j$, défini par convention comme $\tilde{d}_j = \tilde{b}_j - b_j$ [13]. Les coefficients a_{ij}, b_j et c_i sont notés dans le tableau de Butcher

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \\ & \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_{s-1} & \tilde{b}_s \\ \hline & \tilde{d}_1 & \tilde{d}_2 & \cdots & \tilde{d}_{s-1} & \tilde{d}_s \end{array} \tag{II.48}$$

Le but d'une méthode adaptative est de nous permettre de voir si notre simulation est correcte ou non en terme de précision de la solution. Mais il faut faire attention à un point très important. Si l'erreur entre les deux solutions aux deux ordres reste raisonnable par rapport à la tolérance imposée, cela ne veut en aucun cas dire que ces deux solutions sont correctes. En effet, si ces dernières s'éloignent de la solution exacte en même temps, l'erreur entre elles sera sous le seuil de la tolérance mais chacune individuellement sera fausse vis à vis de la solution exacte.
De plus, selon les systèmes simulés, faire une avance temporelle avec un pas de temps constant, défini à l'initialisation de la simulation, ne sera pas efficace voire même contre-productif. Par exemple, si nous étudions un système à deux corps, lorsqu'ils sont loins l'un de l'autre nous pouvons prendre un pas de temps Δt très grand, mais par contre, lorsque les deux corps se rapprochent et interagissent, pour pouvoir capturer toute la physique sous-jacente il faut absolument ralentir la simulation en diminuant le pas de temps. Pour ce faire, nous utiliserons les propriétés intrinsèques de ces méthodes de Runge-Kutta (voir à la section 2.3).
Le fait de calculer deux solutions à des ordres différents, avec les mêmes fonctions auxiliaires k_i mais avec des combinaisons linéaires différentes entre elles, se nomme une méthode *emboîtée* (« Embedded Runge-Kutta »). Il existe différentes méthodes emboîtées. Nous pouvons citer les méthodes de Bogacki et Shampine (ordres 3 et 2), de Felhberg (ordres 5 et 4), de Cash et Karp (ordres 5 et 4), de Dormand et Prince (ordres 5 et 4) et de Verner (ordres 9 et 8).

2.2 Méthodes de Runge-Kutta utilisées

Dans cette section nous présentons les trois méthodes emboîtées de Runge-Kutta qui seront implémentées tout au long de notre étude, puisqu'avec elles nous pouvons utiliser un pas de temps adaptatif efficace ainsi qu'une approximation de la solution à n'importe quel temps d'une boucle de calcul.

2.2.a *Méthode de Dormand et Prince : Runge-Kutta 5(4)7M = RK5(4)*

Nous allons utiliser l'algorithme de Runge-Kutta à l'ordre 5 qui a été introduit par Dormand et Prince [27]. D'après le théorème « Butcher Barriers » ([41] page 173), il n'existe pas de méthode de Runge-Kutta explicite d'ordre $p \geq 5$ pour laquelle nous ayons l'égalité entre le nombre d'étages de calculs et l'ordre de la méthode, tel que $s = p$. Ainsi, sur la base du modèle (II.41), il faut ici, au moins, $s = 6$ étages de calculs si nous voulons une résolution à l'ordre $p = 5$. Les meilleurs coefficients pour minimiser l'erreur (II.43) sont donnés dans le tableau de Butcher en (II.50) et le détail des étapes de calculs de la méthode est le suivant

$$
\begin{aligned}
k_1 &= f\left(t_n, y_n\right) \\
k_2 &= f\left(t_n + c_2\Delta t, y_n + \Delta t a_{21} k_1\right) \\
k_3 &= f\left(t_n + c_3\Delta t, y_n + \Delta t\left(a_{31}k_1 + a_{32}k_2\right)\right) \\
k_4 &= f\left(t_n + c_4\Delta t, y_n + \Delta t\left(a_{41}k_1 + a_{42}k_2 + a_{43}k_3\right)\right) \\
k_5 &= f\left(t_n + c_5\Delta t, y_n + \Delta t\left(a_{51}k_1 + a_{52}k_2 + a_{53}k_3 + a_{54}k_4\right)\right) \\
k_6 &= f\left(t_n + c_6\Delta t, y_n + \Delta t\left(a_{61}k_1 + a_{62}k_2 + a_{63}k_3 + a_{64}k_4 + a_{65}k_5\right)\right) \\
k_7 &= f\left(t_n + c_7\Delta t, y_n + \Delta t\left(a_{71}k_1 + a_{72}k_2 + a_{73}k_3 + a_{74}k_4 + a_{75}k_5 + a_{76}k_6\right)\right) \\
y_{n+1} &= y_n + \Delta t\left(b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4 + b_5k_5 + b_6k_6\right) \\
\tilde{y}_{n+1} &= y_n + \Delta t\left(\tilde{b}_1k_1 + \tilde{b}_2k_2 + \tilde{b}_3k_3 + \tilde{b}_4k_4 + \tilde{b}_5k_5 + \tilde{b}_6k_6 + \tilde{b}_7k_7\right) \\
y_{n+1} - \tilde{y}_{n+1} &= \Delta t\left(\left(b_1 - \tilde{b}_1\right)k_1 + \cdots + \left(b_7 - \tilde{b}_7\right)k_7\right) = -\Delta t\left(\tilde{d}_1k_1 + \cdots + \tilde{d}_7k_7\right)
\end{aligned}
\tag{II.49}
$$

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hline
y_{n+1} & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
\tilde{y}_{n+1} & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40} \\
\hline
 & -\frac{71}{57600} & 0 & \frac{71}{16695} & -\frac{71}{1920} & \frac{17253}{339200} & -\frac{22}{525} & \frac{1}{40}
\end{array}
\tag{II.50}
$$

La présence du vecteur $\tilde{b}$ nous sert à calculer la solution $\tilde{y}_{n+1}$ à l'ordre inférieur $\tilde{p} = p - 1 = 4$. Dormand et Prince ont aussi pris en compte le $FSAL$ (« First Same As Last »). Cela nous permet d'obtenir que la dernière composante du vecteur b soit nulle et que la dernière ligne des a_{ij} soit identique à ce vecteur. Donc, même si nous voyons l'apparition d'un septième étage de calcul, en fait il n'y en a que six, car l'évaluation de cette septième étape peut resservir au premier étage du pas de calcul suivant. Ainsi, il n'y a qu'au tout premier pas de temps de la simulation où il faut sept étages de calcul. Les autres fois, nous remplaçons tout simplement la valeur de k_1 du pas de calcul n par l'ancienne valeur de k_7 du pas de calcul précédent $n - 1$.

En résumé, nous avons donc le système (II.49) pour résoudre notre problème avec $p = 5$, l'ordre de la méthode donnant la solution y_{n+1}, et $\tilde{p} = 4$, l'ordre de la méthode donnant la solution $\tilde{y}_{n+1}$. Le tout se fait avec sept étages de calculs (ou 6 en prenons en compte le $FSAL$). Cela explique la provenance des différents chiffres dans la notation de la méthode de Dormand et Prince, à savoir, *RK5(4)7M* (connue aussi sous le nom de *Dopri5*). Enfin, la lettre M signifie que cette résolution est celle qui a le terme d'erreur le plus optimal par rapport à, par exemple, la méthode notée S qui a une meilleure stabilité.

2.2.b *Méthode de Bogacki et Shampine : Runge-Kutta 3(2)*

Pour notre travail nous utilisons aussi une méthode d'ordre moins élevé. Il existe une méthode de Runge-Kutta emboîtée permettant d'approcher la solution y_{n+1} à l'ordre 3 et d'avoir une autre approximation $\tilde{y}_{n+1}$ à l'ordre 2. Cette méthode dite de *Bogacki et Shampine* porte le nom

de ses deux inventeurs [7]. Nous avons les étapes de calculs

$$\begin{aligned}
k_1 &= f\left(t_n, y_n\right) \\
k_2 &= f\left(t_n + c_2\Delta t, y_n + \Delta t a_{21} k_1\right) \\
k_3 &= f\left(t_n + c_3\Delta t, y_n + \Delta t\left(a_{31}k_1 + a_{32}k_2\right)\right) \\
k_4 &= f\left(t_n + c_4\Delta t, y_n + \Delta t\left(a_{41}k_1 + a_{42}k_2 + a_{43}k_3\right)\right) \\
y_{n+1} &= y_n + \Delta t\left(b_1k_1 + b_2k_2 + b_3k_3\right) \\
\tilde{y}_{n+1} &= y_n + \Delta t\left(\tilde{b}_1k_1 + \tilde{b}_2k_2 + \tilde{b}_3k_3\right) \\
\left|y_{n+1} - \tilde{y}_{n+1}\right| &= \Delta t\left|\tilde{d}_1k_1 + \tilde{d}_2k_2 + \tilde{d}_3k_3 + \tilde{d}_4k_4\right|
\end{aligned} \tag{II.51}$$

avec les différents coefficients

$$\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
3/4 & 0 & 3/4 & & \\
1 & 2/9 & 1/3 & 4/9 & 0 \\
-- & --- & --- & --- & --- \\
y_{n+1} & 2/9 & 1/3 & 4/9 & 0 \\
\tilde{y}_{n+1} & 7/24 & 1/4 & 1/3 & 1/8 \\
-- & --- & --- & --- & --- \\
 & -5/72 & 1/12 & 1/9 & -1/8
\end{array} \tag{II.52}$$

Pour information, cette méthode utilise aussi le *FSAL*, c'est-à-dire que nous avons $k_4^{[n]} = k_1^{[n+1]}$.

2.2.c *Méthode de Verner : Runge-Kutta 16 :9(8)*

Enfin, nous avons besoin d'avoir une méthode temporelle d'ordre très élevé pour réaliser une étude *complète*. Pour cela, nous utilisons l'approche de Runge-Kutta 16 : 9(8) de Verner [76]. Il y a 16 étages de calculs pour obtenir deux solutions, l'une d'ordre 9 et l'autre d'ordre 8. Verner a aussi développé deux autres approches de cette méthode, l'une étant *plus efficace*, l'autre étant *plus robuste*. Puisque nous faisons des méthodes très précises, nous utiliserons la dernière qui est disponible à la référence [1].

2.3 Pas de temps variable

Pour une question d'efficacité, il est préférable de faire un calcul à pas de temps Δt variable. Cela améliore la stabilité des intégrateurs temporels grâce à une estimation de l'erreur locale. Lorsque nous simulons des équations numériquement raides, utiliser un pas de temps constant est totalement inefficace, puisque nous raterions à tous les coups des phénomènes physiques importants en allant trop vite à des endroits où il faudrait ralentir la simulation pour bien cerner le problème. Ou alors le temps de simulation serait beaucoup trop important en prenant un pas de temps très petit pour éviter les erreurs numériques. Il faut donc utiliser un pas de temps adaptatif, mais aussi en choisir un de qualité, pour, par exemple, ne pas perdre un temps

de calcul important sur cette étape. Comme l'a dit Ceschino [17], « D'ordinaire, on se contente de multiplier ou de diviser par 2 la valeur du pas », ce qui n'est pas du tout adapté à des simulations non linéaires. En effet, ce type de technique adaptative nécessite de calculer deux fois le terme non linéaire avec deux pas de temps différents (en général, le rapport entre ces deux pas de temps est pris égal à 2). Ensuite, pour choisir la simulation à garder entre ces deux, il faut comparer les erreurs numériques par rapport à une certaine tolérance. Mais en faisant ainsi, nous réalisons deux fois plus de calculs, ce qui est contre-productif en terme de temps de calcul.

2.3.a *Méthode du I Step Control*

Le plus important est donc de bien choisir la méthode de calcul du pas de temps variable. Nous allons utiliser la méthode de Ceschino qui est aussi appelé le *Integral Step Controller* [41]. C'est une estimation du pas de temps idéal qui comprend une sécurité numérique, pour éviter de choisir un mauvais pas de temps même si « tout semble correct ».

A chaque pas de temps t_n, nous venons de voir que le programme informatique calcule deux approximations y_{n+1} et $\tilde{y}_{n+1}$ de la solution pour arriver au temps t_{n+1}. Une estimation de l'erreur locale est $y_{n+1} - \tilde{y}_{n+1}$. Nous désirons que cette erreur satisfasse la relation

$$|y_{n+1_i} - \tilde{y}_{n+1_i}| \leq \mathrm{sc}_i, \qquad \mathrm{sc}_i = \mathrm{Atol}_i + \max\left(|y_{n_i}|, |y_{n+1_i}|\right) \cdot \mathrm{Rtol}_i \tag{II.53}$$

où Atol et Rtol sont les tolérances absolue et relative et donc sc_i est un mélange de ces types de tolérances. Nous prendrons toujours Atol identique à Rtol.

Comme estimation de l'erreur au pas n, nous faisons appel à la définition

$$\mathrm{err}_n = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_{n+1_i} - \tilde{y}_{n+1_i}}{\mathrm{sc}_i}\right)^2} \tag{II.54}$$

tout en sachant qu'il est possible de choisir d'autres normes.

A ce stade, nous comparons l'erreur err_n avec l'unité 1 pour trouver le pas de temps Δt_{n+1} suivant, c'est-à-dire que nous ne validerons le calcul pour avancer la solution en t_{n+1} uniquement si la condition $\mathrm{err}_n \leq 1$ est respectée. Sinon, le calcul sera dit *rejeté*, c'est-à-dire qu'il sera recommencé au même temps initial mais avec un nouveau pas de temps plus petit. Nous aurons donc fait une boucle de calculs inutile et perdu du temps.

Nous savons que $\mathrm{err}_n = y_{n+1} - \tilde{y}_{n+1} = \left(y_{n+1} - y\left(t_n + \Delta t_n\right)\right) + \left(y\left(t_n + \Delta t_n\right) - \tilde{y}_{n+1}\right) = \mathcal{O}\left(\Delta t_n^{p+1}\right) + \mathcal{O}\left(\Delta t_n^{\tilde{p}+1}\right) \approx C\Delta t_n^{q+1}$ avec $q = \min\left(\tilde{p}, p\right)$ et que $1 \approx C\Delta t_{n+1}^{q+1}$. Nous pouvons donc estimer que $C = \dfrac{1}{\Delta t_{n+1}^{q+1}}$ et nous en déduisons la formulation du nouveau pas de temps Δt_{n+1}, à savoir :

$$\Delta t_{n+1} = \Delta t_n \left(\frac{1}{\mathrm{err}_n}\right)^{\frac{1}{q+1}} \tag{II.55}$$

Afin d'éviter d'augmenter le pas de temps Δt de manière trop importante, à ce calcul se rajoutent des *sécurités numériques* afin d'obtenir le pas de temps optimal

$$\Delta t_{\mathrm{opt}} = \Delta t_n \cdot \min\left(\mathsf{facmax}, \mathsf{fac} \cdot \left(\frac{1}{\mathrm{err}_n}\right)^{\frac{1}{q+1}}\right) \tag{II.56}$$

où facmax est un facteur limitant la croissance de deux pas de temps successifs, généralement compris entre 1.5 et 5 [41]. Il est aussi conseillé de poser facmax $= 1$ juste après un pas de temps rejeté [67] pour éviter de refaire le calcul avec un Δt plus grand que la valeur pour laquelle le calcul vient juste d'échouer.
Au cas où le nouveau pas de temps serait surestimé, nous pouvons réduire cette estimation à l'aide du pré-facteur fac, qui, selon les choix faits dans la littérature, prend une valeur fixe entre 0.8 et 0.9, ou une valeur dépendante de l'ordre de la méthode de Runge-Kutta : $(0.25)^{1/(q+1)}$ ou $(0.38)^{1/(q+1)}$. L'idée sous-jacente est qu'il vaux mieux faire un ou deux calculs supplémentaires pour faire évoluer le pas de temps, plutôt que de risquer d'accumuler des erreurs numériques et de ne pas satisfaire la condition $\mathrm{err}_n \leq 1$, ce qui imposerait un rejet du calcul.
Il faut noter que si le pas de temps doit décroître lors de calculs plus difficiles à résoudre localement, nous ne devons pas le limiter dans sa décroissance pour ne pas accumuler des erreurs de calculs relatives à un pas de temps trop grand.

Applications

- Avec la méthode de Bogacki et Shampine nous avons $q = \min(\tilde{p}, p) = \min(2, 3) = 2$.
- Avec la méthode de Dormand et Prince nous avons $q = \min(\tilde{p}, p) = \min(4, 5) = 4$.
- Avec la méthode de Verner nous avons $q = \min(\tilde{p}, p) = \min(8, 9) = 8$.

2.3.b *Méthode du PI Step Control*

Une alternative pour le calcul du pas de temps variable est le *Proportional Integral Step Control* qui est défini comme ce qui suit d'après [42]. Ici, il faut prendre en compte Δt_n et l'erreur err_n au pas actuel, mais en plus l'erreur au pas de temps précédent notée err_{n-1}. Ce calcul se fait donc en utilisant l'erreur sur deux boucles de calculs successives. Il y a donc une sorte de *mémoire du passé*, cela afin de minimiser le nombre de rejets de boucles de calculs pour cause de pas de temps mal estimé. Cette méthode se met sous la forme

$$\begin{aligned} \Delta t_{n+1} &= \Delta t_n \cdot \left(\frac{1}{\mathrm{err}_n}\right)^{\alpha} \left(\frac{\mathrm{err}_{n-1}}{1}\right)^{\beta} \\ \mathrm{sc}_i &= \mathrm{Atol}_i + \max\left(|y_{n_i}|, |y_{n+1_i}|\right) \cdot \mathrm{Rtol}_i \\ \mathrm{err}_n &\leq 1 \end{aligned} \tag{II.57}$$

Nous pouvons aussi créer le pas de temps optimal

$$\Delta t_{\mathrm{opt}} = \Delta t_n \cdot \min\left(\mathsf{facmax}, \mathsf{fac} \cdot \left(\frac{1}{\mathrm{err}_n}\right)^{\alpha} \left(\frac{\mathrm{err}_{n-1}}{1}\right)^{\beta}\right) \tag{II.58}$$

en utilisant la même démarche que pour (II.56) afin d'obtenir une meilleure *sécurité numérique*. Nous pouvons remarquer que si $\alpha = \dfrac{1}{q+1}$ et $\beta = 0$ nous retombons exactement sur le pas variable précédent. Gustafsson [40] a montré que prendre $\alpha = \dfrac{0.7}{q+1}$ et $\beta = \dfrac{0.4}{q+1}$ était un choix optimal. C'est cette méthode de calcul de pas de temps adaptatif que nous implémentons dans nos codes numériques. De plus, nous choisissons de prendre les tolérances absolue et relative identiques comme cela est souvent le cas dans la littérature [42]. Elles seront nommées simplement *Tolérance* dans la suite de ce manuscrit.

Procédure d'application du pas de temps adaptatif

Le *PI Step Control* est implémenté de la manière suivante :

• Nous partons de la solution connue y_n au temps t_n que nous souhaitons faire évoluer.

• Nous obtenons les solutions y_{n+1} et $\tilde{y}_{n+1}$ d'ordres p et $\tilde{p}$ au temps $t_{n+1} = t_n + \Delta t$, avec un algorithme de type Runge-Kutta. Δt étant le pas de temps défini lors de la boucle de calcul précédente (ou lors de l'initialisation pour le tout premier calcul).

• Nous estimons l'erreur locale err_n qui dépend de la tolérance prédéfinie lors de l'initialisation.

• Nous calculons le nouveau pas de temps optimal Δt_{opt}.

• **IF**$(\text{err}_n \leq 1)$ **THEN**

L'erreur est acceptable, donc nous validons la boucle de calcul.

Nous incrémentons le temps t_n en posant $t_n = t_{n+1}$.

Nous modifions la variable de départ y_n avec la nouvelle valeur y_{n+1} pour continuer l'évolution temporelle.

ELSE

L'erreur n'est pas acceptable, donc nous ne validons pas la boucle de temps.

Nous n'incrémentons pas le temps t_n.

Nous utilisons la valeur précédente y_n pour recalculer la même boucle de temps.

END

• Nous recommençons la procédure avec les nouvelles quantités et le pas de temps optimal Δt_{opt}.

Chapitre III

Intégrateurs exponentiels

Dans ce chapitre, nous faisons un tour d'horizon des méthodes d'intégrateurs exponentiels existantes. Cet état de l'art est basé sur différents travaux bibliographiques, à savoir l'article de revue d'Hochbruck et Ostermann [46], la thèse et l'article de Minchev [55, 56], ainsi que les présentations de Wright [81] et de Berland [5].

1 Présentation

1.1 Pourquoi ?

De manière générique, nous avons une équation aux dérivées partielles de la forme

$$u_t = f(t, u(t)) \tag{III.1}$$

avec la notation $u_t = \dfrac{\partial u}{\partial t}$, que nous séparons en deux parties. L'approche la plus évidente étant de séparer les parties linéaire et non linéaire

$$u_t = \mathscr{L}u + \mathscr{N}(u, t) \tag{III.2}$$

où $\mathscr{L}$ et $\mathscr{N}$ sont respectivement les opérateurs linéaires et non linéaires. La solution étant de la forme $u(x, t)$, avec $u(x, 0) = u_0(x)$. Cette équation aux dérivées partielles peut devenir l'équation aux dérivées ordinaires

$$u_t = \mathbb{L}u + \mathrm{N}(u, t) = f(t, u(t)) \tag{III.3}$$

avec $\mathbb{L}$ la matrice de la partie linéaire. Ce choix n'est bien évidemment pas unique et nous prenons $\mathbb{L}$ autonome, arbitrairement, pour des raisons d'implémentation numérique.

L'idée est de résoudre la partie linéaire de manière exacte et d'approximer au mieux la partie non linéaire de manière explicite. Cette approche est issue des travaux de Certaine [16] qui a ainsi développé le premier intégrateur exponentiel en 1960, ce qui fut une avancée significative par rapport aux méthodes existantes. En effet, comme expliqué dans le chapitre précédent, numériquement il existe deux classes de méthodes, celles qui sont *implicites* mais qui sont beaucoup trop chères en temps de calcul, et celles qui sont *explicites*, mais qui nécessitent des pas de temps tellement petits pour limiter les erreurs que cela les rend souvent inefficaces. Il a donc fallu mettre au point des intégrateurs numériques de haute performance pour un faible coût numérique. Un des problèmes rencontrés pour l'adoption de ces nouvelles méthodes a été l'utilisation centrale d'exponentielles de matrices. Cette fonction était loin d'être simple à manipuler numériquement jusqu'à dernièrement, ce qui explique le faible attrait dans la littérature pour ces intégrateurs. Mais grâce à des avancées significatives dans l'approximation numérique d'exponentielles de matrices et de multiplication par des vecteurs, nous assistons à un regain d'intérêt pour ces techniques. Ainsi, une définition possible de cet outil de calcul serait « Un intégrateur exponentiel est une méthode numérique qui implique une fonction exponentielle (ou une fonction s'y reliant) du Jacobien ou d'une approximation du Jacobien » [56].

1.2 Historique

Les premiers travaux relatifs aux intégrateurs exponentiels sont donc ceux de Certaine [16] qui a obtenu deux de ces méthodes en se basant sur des approches temporelles implicites de type *Adams et Moulton*. Son travail entre dans le cadre des méthodes dites *Exponential Time Differencing* (*ETD*). Quelques années plus tard, Nørsett [59] a fait de même pour des *ETD* basées cette fois sur une approche explicite d'*Adams et Bashforth*, puis Verwer et Van der Houven [75] ont développé une modification qui stabilise ces méthodes en obtenant des *méthodes ETD linéaires à plusieurs pas*.

Friedli [34] a de son côté mis au point des méthodes *ETD* basées sur celles de Runge-Kutta explicites, qui sont très proches des méthodes dites *W* de Steihaug et Wolfbrandt [68] (et de leur extension *Adaptive Runge-Kutta* de Strehmel et Weiner [69]). Ces dernières diffèrent de la méthode de Friedli par le fait que les fonctions utilisées ne sont plus calculées exactement mais sont remplacées par des approximations de Padé.

Cox et Matthews [23] ont quant à eux mis au point des méthodes *ETD* à base de Runge-Kutta pour l'évolution temporelle à ordres élevés (notées *ETDRK*), dont la plus connue, *ETDRK4*, est basée sur la méthode de Runge-Kutta d'ordre 4. L'utilisation de certaines fonctions nécessaires à la résolution numérique est par contre plus compliquée. Ces techniques furent redécouvertes par Beylkin, Keiser et Vozovoi [6] sous le nom *Exact Linear Part*.

Une autre approche majeure, le *Facteur Intégrant*, est due à Lawson [51]. Krogstad [50] a généralisé cette approche avec la méthode *Generalized Integrating Factor*, qui apporte une meilleure précision au regard des méthodes *ETD* et *IF*. Ce travail rentre dans le cadre plus général des méthodes de *Runge-Kutta Munthe-Kaas* [58] qui transforment l'équation différentielle originale en une nouvelle équation évoluant sur des algèbres de Lie.

Toutes ces méthodes sont divisées en trois grandes classes d'intégrateurs numériques : les méthodes *Linéaires à plusieurs pas* (ou *multivaluées*), les méthodes *à plusieurs étages* (de type Runge-Kutta) et les méthodes hybrides *Générales Linéaires*.

2 Méthodes Exponentielles Linéaires multivaluées

Dans la classe des intégrateurs exponentiels, nous distinguons les méthodes dites *Exponential Time Differencing*, notées *ETD*, et les *Integrating Factor* (ou *Facteur Intégrant*), notées *IF*. Dans la littérature, les méthodes les plus fréquemment utilisées sont les *ETD*, particulièrement en Physique [47, 63, 72]. Mais dans notre étude, nous nous intéresserons plutôt à l'autre approche, le facteur intégrant.

2.1 Exponential Time Differencing (ETD)

Comme ce sont ces méthodes qui sont les plus courantes, elles ont souvent été (re)découvertes de manière indépendante et sous différents noms, tels que « Generalized Linear Multistep method » et « Generalized Runge-Kutta method » [75, 78], « Exact Linear Part » [6] ou encore « Exponential propagation » [30, 35]. Nous devons le nom final *ETD* à Cox et Matthews [23].

2.1.a *Formule exacte d'intégration*

L'idée avec cette approche est de construire un intégrateur exponentiel à l'aide de la formule de la variation des constantes. Nous partons de l'équation générique des problèmes que nous considérons (III.3) et nous la multiplions par le terme $\mathrm{e}^{-\mathbb{L}t}$ pour obtenir

$$\mathrm{e}^{-\mathbb{L}t}u_t - \mathrm{e}^{-\mathbb{L}t}\mathbb{L}u = \mathrm{e}^{-\mathbb{L}t}\mathrm{N}(u,t) \tag{III.4}$$

Nous intégrons sur un pas de temps de calcul Δt entre t_n et $t_n + \Delta t$, pour obtenir la relation exacte, avec $u_n = u(t_n)$,

$$\left(\mathrm{e}^{-\mathbb{L}\Delta t}u_{n+1} - u_n\right)\mathrm{e}^{-\mathbb{L}t_n} = \int_{t_n}^{t_n+\Delta t} \mathrm{e}^{-\mathbb{L}t}\mathrm{N}(u,t)\mathrm{d}t \tag{III.5}$$

ou de manière équivalente

$$u_{n+1} = \mathrm{e}^{\mathbb{L}\Delta t}u_n + \mathrm{e}^{\mathbb{L}\Delta t}\int_0^{\Delta t} \mathrm{e}^{-\mathbb{L}\tau}\mathrm{N}\left(u\left(x, t_n+\tau\right), t_n+\tau\right)\mathrm{d}\tau \tag{III.6}$$

La variété des méthodes *ETD* vient de la manière dont est approximée l'intégrale contenant le terme non linéaire dans l'expression (III.6), ce qui peut être fait de nombreuses façons comme nous allons le voir.

2.1.b *Solution exacte*

La solution exacte est de la forme

$$u_{n+1} = \mathrm{e}^{\mathbb{L}\Delta t}u_n + \sum_{l=0}^{\infty} \varphi^{[l+1]}(\mathbb{L}\Delta t)\Delta t^{l+1}N_n^{(l)} \tag{III.7}$$

avec $N_n^{(i)} = \left.\dfrac{\mathrm{d}^i}{\mathrm{d}t^i}\right|_{t=t_n} N(u(t),t)$ et les fonctions φ sont définies par les relations

$$\begin{aligned}
\varphi^{[l]} &= \varphi^{[l]}(\Delta t\ \mathbb{L}) \\
\varphi^{[l,\lambda]} &= \varphi^{[l]}(\lambda\Delta t\ \mathbb{L}) = \frac{1}{\lambda^l h^l}\int_0^{\lambda\Delta t} \mathrm{e}^{(\lambda\Delta t-\tau)\mathbb{L}}\frac{\tau^{l-1}}{(l-1)!}\mathrm{d}\tau \\
\varphi^{[l,k]} &= \varphi^{[l]}(c_k\Delta t\ \mathbb{L})
\end{aligned} \tag{III.8}$$

qui sont évaluées à la précision machine, à l'aide d'approximations de Padé à ordre élevé et d'autres techniques. Ces fonctions φ doivent satisfaire la relation de récurrence

$$\varphi^{[0]}(z) = \mathrm{e}^z, \quad \varphi^{[l]}(z) = \frac{1}{l!}, \quad \varphi^{[l+1]}(z) = \frac{\varphi^{[l]}(z) - \frac{1}{l!}}{z}, \quad l = 0, 1, 2, \cdots \tag{III.9}$$

Comme énoncé dans la définition d'un intégrateur exponentiel en III.1.1, nous voyons qu'au lieu d'utiliser directement la fonction exponentielle, ici nous faisons appel à des fonctions intermédiaires φ.

Pour construire une approximation polynomiale de $N(u,t)$ en intégrant de t_n à t_{n+1} il existe trois approches :

- Utiliser des étages de calculs intermédiaires, c'est-à-dire trouver des approximations d'ordre peu élevé U_i de u à différents points compris entre $t_n < t < t_{n+1}$ et utiliser $N(U_i)$ dans une sorte de loi de quadrature. Cela définit une approche de type Runge-Kutta.
- Utiliser les valeurs de u à des temps précédents. C'est l'idée des schémas à plusieurs pas de temps comme les méthodes d'Adams et Bashforth (par exemple, la méthode *ETD2*).
- Combiner les deux approches ci-dessus. Cela nous donne une nouvelle classe de technique, les méthodes linéaires générales, dont nous présenterons la formulation générale en III.4.

2.1.c *Approximations de l'intégrale*

Nous pouvons remplacer le terme non linéaire N par une interpolation polynomiale et résoudre exactement l'intégrale qui en résulte.

Exemple : Méthode d'ordre 1

• La première idée est d'approximer $\mathrm{N}\left(u\left(x, t_n+\tau\right), t_n+\tau\right)$ par une constante N_n, ce qui nous donne

$$\begin{aligned}
u_{t_n+1} &= \mathrm{e}^{\mathbb{L}\Delta t}u_{t_n} + \mathrm{e}^{\mathbb{L}\Delta t}\int_0^{\Delta t}\mathrm{e}^{-\mathbb{L}\tau}\mathrm{N}_n\mathrm{d}\tau \\
&= \mathrm{e}^{\mathbb{L}\Delta t}u_{t_n} + \varphi^{[1]}(\mathbb{L}\Delta t)\Delta t N_n
\end{aligned} \tag{III.10}$$

avec certaines fonctions φ définies par

$$\varphi^{[1]}(z) = \frac{\mathrm{e}^z - 1}{z} \tag{III.11}$$

Cette méthode, qui se réduit à un schéma d'Euler classique pour $\mathbb{L} = 0$, est connue sous le nom d'*ETD-Euler* ou encore *ETD1*.

• De la même manière, nous pouvons obtenir le schéma *ETD Euler implicite*

$$u_{t_{n+1}} = \mathrm{e}^{\mathbb{L}\Delta t}u_{t_n} + \varphi^{[1]}(\mathbb{L}\Delta t)\Delta t N_{n+1} \tag{III.12}$$

Exemple : Méthode d'ordre 2

• En prenant une approximation d'ordre supérieur, nous pouvons construire le schéma *ETD2*

$$\begin{aligned} N &= N_n + \tau \frac{N_n - N_{n-1}}{\Delta t} + O(\Delta t^2) \\ u_{t_{n+1}} &= \mathrm{e}^{\mathbb{L}\Delta t} u_{t_n} + \varphi^{[1]}(\mathbb{L}\Delta t)\Delta t N_n + \varphi^{[2]}(\mathbb{L}\Delta t)\Delta t N_{n-1} \end{aligned} \tag{III.13}$$

avec cette fois les fonctions φ suivantes

$$\begin{aligned} \varphi^{[1]}(z) &= \frac{(1+z)\mathrm{e}^z - 1 - 2z}{z^2} \\ \varphi^{[2]}(z) &= \frac{1 + z - \mathrm{e}^z}{z^2} \end{aligned} \tag{III.14}$$

2.2 Facteur Intégrant (*IF*)

Le *facteur intégrant* est une méthode dont l'emploi d'une fonction exponentielle permet un changement de variable afin de faciliter la résolution numérique d'équations d'évolutions. Nous devons cette approche, différente de celle des *ETD* pour la construction d'une méthode exponentielle, à Lawson [51]. Ici, à la place d'intégrer sur un pas de temps, nous allons faire un changement de variable. Nous avons la relation de départ

$$u'(t) - \mathbb{L}u(t) = \mathrm{N}(t, u(t)) \tag{III.15}$$

Posons le changement de variable

$$u(t) = \mathrm{e}^{L(t-t_n)} v(t) \tag{III.16}$$

Si nous dérivons cette relation, nous pouvons la reécrire sous la forme

$$u'(t) = \mathbb{L}\ \mathrm{e}^{\mathbb{L}(t-t_n)} v(t) + \mathrm{e}^{\mathbb{L}(t-t_n)} v'(t) \tag{III.17}$$

ou encore comme

$$u'(t) - \mathbb{L}u(t) = \mathrm{e}^{\mathbb{L}(t-t_n)} v'(t) \tag{III.18}$$

Nous obtenons l'équation d'évolution non plus dans la variable initiale u, mais dans la nouvelle variable v

$$\begin{aligned} v'(t) &= \mathrm{e}^{-\mathbb{L}(t-t_n)} \left(u'(t) - \mathbb{L}u(t)\right) \\ &= \mathrm{e}^{-\mathbb{L}(t-t_n)} \mathrm{N}(t, u(t)) \end{aligned} \tag{III.19}$$

Ainsi, nous avons réussi à retirer le terme linéaire de l'équation différentielle. Le fait d'avoir une exponentielle décroissante de la matrice des termes linéaires va nous permettre de réduire une partie de la raideur numérique et des oscillations [11]. Dans ce manuscrit, nous appelons cette méthode le *facteur intégrant classique*.

Choix de la matrice linéaire

Le choix de la matrice des termes linéaires $\mathbb{L}$ dépend de l'équation étudiée. Par exemple, considérons l'équation $y'(t) = \sin(y)$. D'après le développement du sinus en y proche de zéro, la partie linéaire est tout simplement y. Donc notre équation va devenir $y'(t) - y(t) = \sin(y) - y(t)$. Le second membre n'est donc plus composé que de la partie non linéaire.

2.2.a *Aspect pratique de l'implémentation*

D'un point de vue pratique, deux options s'offrent à nous :
- Soit nous pouvons passer de u à v au temps initial de la simulation, tout calculer en v puis revenir en u au temps final de la simulation.
- Soit nous pouvons translater le temps à chaque boucle de calcul, c'est à dire que la valeur de t_n change à chaque pas de temps.

Dans ce dernier cas, à chaque pas de calcul nous passons de u à v en t_n, nous calculons l'évolution en v sur un pas de temps, puis en $t_{n+1} = t_n + \Delta t$ nous revenons en u. Nous appliquerons donc cette deuxième méthode, en translatant le temps t_n, puisque c'est elle qui minimise l'incertitude de calcul. En effet, nous passons de u à v et vice versa plus souvent que dans le premier cas, ce qui a pour incidence de minimiser les erreurs de calcul. Pour comprendre cela nous pouvons par exemple faire calculer $\sin(\pi)$ à *Matlab*, puis d'autres valeurs modulo 2π et comparer les résultats. Nous obtenons ainsi $\sin(\pi) = 1.2246\ 10^{-16}$, soit le zéro machine. C'est correct. Mais dès que nous augmentons l'argument périodiquement, nous obtenons de plus en plus d'erreurs numériques. En effet, $\sin(10\ \pi) = -1.2246\ 10^{-15}$, $\sin(10^3\ \pi) = -3.2142\ 10^{-13}$ et $\sin(10^5\ \pi) = -3.3961\ 10^{-11}$. Si nous allons encore plus loin, nous trouvons que $\sin(10^{13}\ \pi) = -0.0027$ et que $\sin(10^{15}\ \pi) = -0.2362$. Nous voyons bien que nous sommes très loin du zéro machine. C'est pour cela qu'il faut toujours calculer ce type de fonctions près de l'origine, d'où le changement de temps t_n à chaque boucle de temps.

2.2.b *Choix de l'avancement temporel*

Une fois ce changement de variable effectué, d'un point de vue numérique l'idée est d'appliquer un intégrateur numérique pour l'évolution temporelle de la nouvelle équation (III.19) (par exemple de type Runge-Kutta pour la méthode de Lawson, voir la section III.3.2), puis de revenir dans la variable originelle une fois le résultat obtenu, grâce à la relation (III.16) entre les deux variables u et v. C'est ce que nous allons voir dans la section suivante.

3 Méthodes Exponentielles à plusieurs étages

Plusieurs évolutions *simples* de la méthode précédente existent, comme le schéma de *Lawson-Euler* ou dit méthode explicite *IF Euler* dans la variable v

$$v_{n+1} = v_n + \Delta t N(v_n) \tag{III.20}$$

ou en revenant à la variable de départ u

$$u_{n+1} = \mathrm{e}^{\mathbb{L}\Delta t} u_n + \mathrm{e}^{\mathbb{L}\Delta t} \Delta t N_n \tag{III.21}$$

avec $v_n = \mathrm{e}^{-\mathbb{L}\Delta t} u_n$ et $N(u_n) = N_n$.

De la même manière, nous pouvons obtenir la méthode implicite d'Euler

$$u_{n+1} = \mathrm{e}^{\mathbb{L}\Delta t} u_n + \mathrm{e}^{\mathbb{L}\Delta t} \Delta t N_{n+1} \tag{III.22}$$

Les méthodes à plusieurs étages de calculs peuvent être aussi plus complexes, que ce soit à base de méthodes de Runge-Kutta, ou à base de méthodes de Rosenbrock [44]. Nous allons uniquement regarder le premier cas qui est le plus utilisé. La première approche du genre est due une fois de plus à Lawson [51]. Après avoir obtenu l'équation (III.19) par un changement de variable, le principe est d'utiliser une méthode de Runge-Kutta pour son évolution temporelle. Ainsi, la méthode de facteur intégrant est directement liée avec la méthode d'avancement temporel de Runge-Kutta. Pour information, même si nous ne regardons pas ce cas ici, il faut savoir qu'il en va de même pour l'approche *ETD*, dont les premières méthodes à base de Runge-Kutta sont à créditer à Friedli [34] et dont la plus connue est celle de Cox et Matthews [23], notée *ETDRK4*.

3.1 Runge-Kutta Exponentiel

- De manière générale, les méthodes à s étages se notent

$$\begin{aligned} U_i &= \sum_{j=1}^{s} \left(\sum_{l=1}^{m} \alpha_{ij}^{[l]} \varphi^{[l]} \right) \Delta t N(U_j) + \chi_i(\mathbb{L}\Delta t) u_{n-1} \\ u_n &= \sum_{i=1}^{s} \left(\sum_{l=1}^{m} \beta_{i}^{[l]} \varphi^{[l]} \right) \Delta t N(U_i) + \chi(\mathbb{L}\Delta t) u_{n-1} \end{aligned} \tag{III.23}$$

avec m qui est la limite du nombre des fonctions $\varphi^{[i]}$, où les U_i sont les étapes internes d'approximation de la solution exacte pour $i = 1, \cdots, s$, avec les fonctions φ (III.8) et avec les fonctions exponentielles standards

$$\chi(z) = \mathrm{e}^{z}, \quad \chi_i(z) = \mathrm{e}^{c_i z}, \quad i = 1, \cdots, s \tag{III.24}$$

- Une autre notation utilisée, introduite par Friedli [34], est

$$\begin{aligned} a_{ij}(z) &= \sum_{l=1}^{s} \alpha_{ij}^{[l]} \varphi^{[l]}(c_i z) \\ b_i(z) &= \sum_{l=1}^{s} \beta_{i}^{[l]} \varphi^{[l]}(z) \end{aligned} \tag{III.25}$$

avec le coefficient $z = \mathbb{L}\Delta t$.

Nous pouvons aussi regrouper les coefficients a_{ij}, b_i et c_i dans un tableau de Butcher étendu

$$\begin{array}{c|ccc|c} c_1 & a_{11}(z) & \cdots & a_{1s}(z) & \chi_1(z) \\ \vdots & \vdots & & \vdots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} & \chi_s(z) \\ -- & -- & -- & -- & -- \\ & b_1(z) & \cdots & b_s(z) & \chi(z) \end{array} \tag{III.26}$$

- Pour le cas limite $\mathbb{L} \to 0$, nous obtenons la méthode de Runge-Kutta sous-jacente.
- Le principe de construction de telles approches nous est expliqué dans la référence [45].

Les deux exemples vus en introduction se présentent alors sous la forme :

Exemple 1 : Méthode de Lawson-Euler

Le tableau suivant représente la méthode de Lawson-Euler $u_{n+1} = \mathrm{e}^z u_{t_n} + \mathrm{e}^z \Delta t N_n$

$$\begin{array}{c|c|c} 0 & 0 & 1 \\ -- & -- & -- \\ & \mathrm{e}^z & \mathrm{e}^z \end{array} \tag{III.27}$$

Exemple 2 : Méthode ETD-Euler

Le tableau suivant représente la méthode d'*ETD-Euler* $u_{n+1} = \mathrm{e}^z u_{t_n} + \varphi^{[1]}(z)\Delta t N_n$

$$\begin{array}{c|c|c} 0 & 0 & 1 \\ -- & -- & -- \\ & \varphi^{[1]}(z) & \mathrm{e}^z \end{array} \tag{III.28}$$

3.2 Cas particulier : *IF RK*

Nous pouvons citer les deux cas particuliers de Runge-Kutta Exponentiels qui sont les plus connus, à savoir les méthodes à base de facteur intégrant, notée *IF RK*, et les méthodes à base d'*ETD*, notée *ETD RK*. La différence entre ces deux approches vient du choix qui est fait pour les fonctions φ. Dans notre étude nous ne regarderons que les approches *IF RK*.
Ces deux techniques sont en fait celles que nous avons précédemment étudiées (*IF* et *ETD*) mais conjuguées à une avance temporelle de type Runge-Kutta. Il est tout à fait possible d'utiliser un facteur intégrant et un Runge-Kutta de manière indépendante, mais il est aussi possible de les lier directement via les coefficients des méthodes, ce qui est fait ici. Ainsi, pour une même technique *IF* et plusieurs méthodes de Runge-Kutta, il est possible de développer de nombreuses méthodes *IF RK* différentes.

Spécificité

Il faut considérer une approximation du terme non linéaire de la forme $N(u(t_n+\tau), t_n+\tau) \approx \delta_i \mathrm{e}^{\mathbb{L}(t_n+\tau)}$, où δ_i est une constante telle que l'approximation soit exacte pour $\tau = c_i\Delta t$. Dans ce cas, pour $l = 1, \cdots, s$ nous avons

$$\varphi^{[l,\lambda]}(\mathbb{L}\Delta t) = \mathrm{e}^{(\lambda - c_l)\mathbb{L}\Delta t} \tag{III.29}$$

De manière générale, les méthodes à s étages *IF RK* ou *Lawson-Runge-Kutta* s'écrivent

$$\begin{aligned} U_i &= \sum_{j=1}^{i=1} a_{ij} \mathrm{e}^{(c_i - c_j)\mathbb{L}\Delta t} \Delta t \mathrm{N}(U_j) + \mathrm{e}^{c_i \mathbb{L}\Delta t} u_{n-1}, \ i = 1, \cdots, s \\ u_n &= \sum_{i=1}^{s} b_i \mathrm{e}^{(1-c_j)\mathbb{L}\Delta t} \Delta t \mathrm{N}(U_i) + \mathrm{e}^{\mathbb{L}\Delta t} u_{n-1} \end{aligned} \tag{III.30}$$

Exemple : Méthode de Lawson

L'intégrateur exponentiel de Lawson, qui est le plus connu, est basé sur la méthode de Runge-Kutta classique d'ordre 4 et est noté *IF RK4*. Dans ce cas, les fonctions φ sont uniquement des exponentielles. Cette méthode est très simple à implémenter et *peu coûteuse* en temps de calcul.

TABLE III.1 – Quatrième ordre de Lawson

0	0	0	0	0	1
$\frac{1}{2}$	$\frac{1}{2}e^{\frac{1}{2}\mathbb{L}\Delta t}$	0	0	0	$e^{\frac{1}{2}\mathbb{L}\Delta t}$
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$e^{\frac{1}{2}\mathbb{L}\Delta t}$
1	0	0	$e^{\frac{1}{2}\mathbb{L}\Delta t}$	0	$e^{\mathbb{L}\Delta t}$
1	$\frac{1}{6}e^{\mathbb{L}\Delta t}$	$\frac{1}{3}e^{\frac{1}{2}\mathbb{L}\Delta t}$	$\frac{1}{3}e^{\frac{1}{2}\mathbb{L}\Delta t}$	$\frac{1}{6}$	$e^{\mathbb{L}\Delta t}$

4 Méthodes Générales Linéaires

Dans les sections III.2 et III.3, nous avons vu respectivement les méthodes linéaires multivaluées et multi-étages. Il existe une généralisation de ces approches qui se nomme les *Méthodes Générales Linéaires*, notée *GLM* [45]. A l'aide de cette généralité, il est possible de développer de nouvelles méthodes plus efficaces que les précédentes.

4.1 Ecriture

Nous partons d'une équation de la forme (III.1). Supposons qu'au début du calcul de chaque pas de temps n, nous connaissons les r quantités suivantes, qui ont été obtenues au pas de calcul précédent $n-1$

$$u_1^{[n-1]}, u_2^{[n-1]}, \cdots, u_r^{[n-1]} \tag{III.31}$$

De même, au pas de calcul n nous devons avoir les quantités

$$u_1^{[n]}, u_2^{[n]}, \cdots, u_r^{[n]} \tag{III.32}$$

Notons U le résultat des calculs intermédiaires (les différents étages de calcul) au pas de temps n

$$U_1, U_2, \cdots, U_s \tag{III.33}$$

et $f(U)$ leurs dérivées

$$f(U_1), f(U_2), \cdots, f(U_s) \tag{III.34}$$

Enfin, toutes ces quantités sont reliées entre elles par le système d'équations

$$\begin{aligned} U_i &= \Delta t \sum_{j=1}^{s} a_{ij}(z) f(U_j) + \sum_{j=1}^{r} w_{ij}(z) u_j^{[n]} \qquad i = 1, \cdots, s \\ u_i^{[n+1]} &= \Delta t \sum_{j=1}^{s} b_{ij}(z) f(U_j) + \sum_{j=1}^{r} v_{ij}(z) u_j^{[n]} \qquad i = 1, \cdots, r \end{aligned} \tag{III.35}$$

Les coefficients sont groupés dans quatre matrices $A(z) = (a_{ij}(z))$, $B(z) = (b_{ij}(z))$, $V(z) = (v_{ij}(z))$ et $W(z) = (w_{ij}(z))$, lesquelles sont mises dans le tableau

$$\begin{array}{c|c|c} c & A(z) & W(z) \\ -- & -- & -- \\ & B(z) & V(z) \end{array} \tag{III.36}$$

Il existe aussi la notation

$$\left[\begin{array}{c} U \\ \hline u^{[n]} \end{array}\right] = \left[\begin{array}{c|c} A & W \\ \hline B & V \end{array}\right] \left[\begin{array}{c} \Delta t f(U) \\ \hline u^{[n-1]} \end{array}\right] \tag{III.37}$$

avec les différents vecteurs

$$U = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_s \end{bmatrix}, \quad f(U) = \begin{bmatrix} f(U_1) \\ f(U_2) \\ \vdots \\ f(U_s) \end{bmatrix}, \quad u^{[n-1]} = \begin{bmatrix} u_1^{[n-1]} \\ u_2^{[n-1]} \\ \vdots \\ u_r^{[n-1]} \end{bmatrix}, \quad u^{[n]} = \begin{bmatrix} u_1^{[n]} \\ u_2^{[n]} \\ \vdots \\ u_r^{[n]} \end{bmatrix} \tag{III.38}$$

Exemple : Méthode de Runge-Kutta d'ordre 4

En utilisant la méthode de Runge-Kutta classique d'ordre 4, nous obtenons le système

$$\left[\begin{array}{c} U_1 \\ U_2 \\ U_3 \\ U_4 \\ \hline u_n \end{array}\right] = \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & 0 & 0 & 1 \\ 0 & \frac{1}{2} & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & 1 \end{array}\right] \left[\begin{array}{c} \Delta t f(U_1) \\ \Delta t f(U_2) \\ \Delta t f(U_3) \\ \Delta t f(U_4) \\ \hline u_{n-1} \end{array}\right] \tag{III.39}$$

4.2 Méthodes Exponentielles Générales Linéaires

Nous pouvons étendre les méthodes générales linéaires dans un système exponentiel

$$\begin{aligned} U_i &= \sum_{j=1}^{s} A_{ij}(\varphi) \Delta t N(u_j) + \sum_{j=1}^{r} W_{ij}(\varphi) u_j^{[n-1]} \qquad i = 1, \cdots, s \\ u_i^{[n]} &= \sum_{j=1}^{s} B_{ij}(\varphi) \Delta t N(u_j) + \sum_{j=1}^{r} V_{ij}(\varphi) u_j^{[n-1]} \qquad i = 1, \cdots, r \end{aligned} \tag{III.40}$$

ou encore sous la forme matricielle

$$\left[\begin{array}{c} U \\ \hline u^{[n]} \end{array}\right] = \left[\begin{array}{c|c} A(\varphi) & W(\varphi) \\ \hline B(\varphi) & V(\varphi) \end{array}\right] \left[\begin{array}{c} \Delta t N(U) \\ \hline u^{[n-1]} \end{array}\right] \tag{III.41}$$

où chaque coefficient des différentes matrices A, B, V et W est relié à une combinaison linéaire des fonctions φ.

Pour information, les méthodes exponentielles de Runge-Kutta étudiées dans la section 3 de ce chapitre sont contenues dans cette méthode générique.

4.2.a *Méthodes de Runge-Kutta Munthe-Kaas*

Les méthodes de Runge-Kutta Munthe-Kaas [58], notées *RKMK*, transforment l'équation différentielle originale en une nouvelle équation évoluant sur des algèbres de Lie (qui est un espace linéaire). Sans rentrer dans les détails (voir [55] page 56), la solution de (III.3) se met sous la forme

$$u(t_n + t) = \mathrm{e}^{tL} u_n + t\varphi^{[1]}(tL) z(t) \tag{III.42}$$

et nous obtenons la nouvelle équation différentielle pour $z(t)$

$$z'(t) = {\varphi^{[1]}}^{-1}(tL) \left(N(\mathrm{Exp}(tL, z) \cdot u_n, t_n + t) - \frac{z}{t} \right) + \frac{z}{t} \tag{III.43}$$

où *Exp* est l'*exponential map* liée à un difféomorphisme noté Ψ. Cette approche a des similitudes avec la technique *IF RK* qui fait qu'il est possible de dire que chaque *IF RK* est une méthode de *RKMK* avec un certain choix du difféomorphisme Ψ [49].

Exemple : Méthode de Runge-Kutta Munthe-Kaas d'ordre 4

Voici le tableau de Butcher pour la méthode à base d'un Runge-Kutta classique d'ordre 4.

TABLE III.2 – Quatrième ordre de Runge-Kutta Munthe-Kaas, *RKMK4*

0	0	0	0	0	1
$\frac{1}{2}$	$\frac{1}{2}\varphi^{[1,2]}$	0	0	0	$\mathrm{e}^{\frac{1}{2}\mathbb{L}\Delta t}$
$\frac{1}{2}$	$\frac{1}{8}\mathbb{L}\Delta t\varphi^{[1,2]}$	$\frac{1}{2}(1 - \frac{1}{4}\mathbb{L}\Delta t)\varphi^{[1,2]}$	0	0	$\mathrm{e}^{\frac{1}{2}\mathbb{L}\Delta t}$
1	0	0	$\varphi^{[1]}$	0	$\mathrm{e}^{\mathbb{L}\Delta t}$
	$\frac{1}{6}(1 + \frac{1}{2}\mathbb{L}\Delta t)\varphi^{[1]}$	$\frac{1}{3}\varphi^{[1]}$	$\frac{1}{3}\varphi^{[1]}$	$\frac{1}{6}(1 - \frac{1}{2}\mathbb{L}\Delta t)\varphi^{[1]}$	$\mathrm{e}^{\mathbb{L}\Delta t}$

4.2.b *Facteur intégrant généralisé*

Se basant sur l'idée précédente, Krogstad a généralisé le facteur intégrant de Lawson [55]. Les explications suivantes sont issues de l'article [50]. L'idée est d'approcher le terme f de (III.1) autour de u_0 grâce à un champ de vecteur modifié $\tilde{f}$, avec $\tilde{f}(u_0, 0) = f(u_0, t_0)$. Le problème de départ peut s'écrire comme

$$\tilde{u}'(\tau) = \tilde{f}(\tilde{u}(\tau), \tau), \qquad \tilde{u}(0) = \tilde{u}_0 \tag{III.44}$$

Avec un opérateur de flux $\varphi^{[\tau,\tilde{f}]}$ tel que $\tilde{u}(\tau) = \varphi^{[\tau,\tilde{f}]}(\tilde{u}_0)$ soit solution de (III.44), de son Jacobien $D\varphi^{[\tau,\tilde{f}]}$ et de $v(\tau)$ avec $v(0) = u_0$, nous obtenons la solution de (III.1) qui s'écrit

$$u_{t_0+\tau} = \varphi^{[\tau,\tilde{f}]}(v(\tau)) \tag{III.45}$$

En différenciant cette relation par rapport à τ, nous obtenons

$$f(u, t_0 + \tau) = \tilde{f}(\varphi^{[\tau,\tilde{f}]}(v), \tau) + D\varphi^{[\tau,\tilde{f}]}(v) v'(\tau) \tag{III.46}$$

ce qui nous donne

$$v'(\tau) = \left(D\varphi^{[\tau,\tilde{f}]}(v) \right)^{-1} \left(f(\varphi^{[\tau,\tilde{f}]}(v), t_0 + \tau) - \tilde{f}(\varphi^{[\tau,\tilde{f}]}(v), \tau) \right) \tag{III.47}$$

Krogstad a observé que $D\varphi^{[\tau,\tilde{f}]}(v) = \mathrm{e}^{\mathbb{L}\tau}$. De ce fait, d'après (III.1), nous voyons bien que si nous prenons $\tilde{f}(u) = \mathbb{L}u$, nous retombons sur le facteur intégrant classique (III.19). De manière plus générale, nous pouvons choisir une approximation polynomiale c du terme non linéaire autour de u_0, telle que $\tilde{f}(u) = \mathbb{L}u + c(\tau)$, avec la relation

$$c(\tau) \approx N(u(t_0 + \tau), t_0 + \tau) \tag{III.48}$$

Cela nous permet de reécrire (III.47) sous la forme

$$v'(\tau) = \mathrm{e}^{-\mathbb{L}\tau}\left(N(u(t_0 + \tau), t_0 + \tau) - c(\tau)\right) \tag{III.49}$$

La question est de savoir quelle approximation choisir. Krogstad a fait le choix d'une approximation polynomiale de Lagrange passant par les k points N_n , N_{n-1} , . . . , N_{n-k+1} pour $k \leqslant n$. En prenant en compte que deux temps de calcul successifs sont reliés par $t_{n+1} = t_n + \Delta t$, donc que $t_n = t_0 + n\Delta t$ pour un pas de temps Δt constant, nous avons plusieurs choix d'ordre (0, 1 et 2) pour $c(\tau)$:

$$\begin{aligned}
c_0(\tau) &= N_n \\
c_1(\tau) &= N_n + \tau\left(\frac{N_n - N_{n-1}}{\Delta t}\right) \\
c_2(\tau) &= N_n + \tau\left(\frac{\frac{1}{2}N_{n-2} - 2N_{n-1} + \frac{3}{2}N_n}{\Delta t}\right) + \tau^2\left(\frac{N_{n-2} - 2N_{n-1} + N_n}{\Delta t}\right)
\end{aligned} \tag{III.50}$$

A partir de maintenant, il suffit de faire comme pour le facteur intégrant de Lawson, c'est-à-dire utiliser une méthode d'avancement temporelle explicite. Krogstad a fait le choix du Runge-Kutta classique d'ordre 4 afin d'obtenir les coefficients des tables III.3 et III.4. Une fois la solution calculée, il nous suffit de revenir dans la variable originelle grâce aux relations suivantes d'ordres différents selon la forme de $c(\tau)$ choisie

$$\begin{aligned}
u(t_n + \tau) &= \mathrm{e}^{\tau\mathbb{L}}v(t) + \tau\varphi^{[1]}(\tau\mathbb{L})N_n \\
u(t_n + \tau) &= \mathrm{e}^{\tau\mathbb{L}}v(t) + \tau\varphi^{[1]}(\tau\mathbb{L})N_n + \frac{\tau^2}{\Delta t}\varphi^{[2]}(\tau\mathbb{L})(N_n - N_{n-1})
\end{aligned} \tag{III.51}$$

De manière plus générale cela donne la relation suivante entre les variables u et v

$$u(t_{n+1}) = \mathrm{e}^{\mathbb{L}\Delta t}v(t_{n+1}) + \sum_{l=1}^{\infty}\varphi^{[l]}(\mathbb{L}\Delta t)\Delta t^l c_k^{(l-1)} \tag{III.52}$$

Facteur intégrant généralisé avec pas de temps variable

Afin de comparer notre méthode de facteur intégrant modifié décrite dans le chapitre suivant avec ce travail, il est nécessaire de reécrire le polynôme c_2 en (III.50), pour tenir compte de l'utilisation d'un pas de temps variable (voir dans le chapitre II, à la section 2.3), comme

$$c_2(\tau) = N_n + \tau\left(\frac{3N_n - 3N_{n-1}}{2\Delta t_n} - \frac{N_{n-1} - N_{n-2}}{2\Delta t_{n-1}}\right) + \tau^2\left(\frac{N_n - N_{n-1}}{\Delta t_n^2} - \frac{N_{n-1} - N_{n-2}}{\Delta t_n \Delta t_{n-1}}\right) \tag{III.53}$$

avec Δt_n et Δt_{n-1} les pas de temps aux boucles de calculs n et $n-1$. Le polynôme $c_1(\tau)$ ne faisant appel qu'à un seul pas de temps $\Delta t = \Delta t_n$ entre deux temps t_{n-1} et t_n, nous ne le modifions pas.

Remarques

Plus nous augmentons l'ordre du polynôme, plus la précision de la solution est augmentée. Krogstad conclut que cette méthode d'ordre 4 est plus précise que celle bien connue de Cox et Matthews [23].

Considérons $k = 1, 2, 3$, tel que les trois choix c_0, c_1 et c_2 du polynôme soient reliés à k par la notation c_{k-1}. Krogstad a nommé ces méthodes *ETDk/RK4* puisque le flux du vecteur champ $\tilde{f} = Lu + c_{k-1}(\tau)$ est égal à la méthode *ETDk* pour $\tau = \Delta t$ et qu'elles utilisent un intégrateur Runge-Kutta d'ordre 4. De manière plus générale il les nomme *GIF* pour *Facteur Intégrant Généralisé*. Krogstad a fait ce choix car il ignorait l'existence de l'article original de Lawson. Plus tard, Minchev [56] a suggéré de plutôt nommer ces méthodes *Lawson Généralisé*, notées *GL*, pour un retour aux sources.

Il est à noter que pour les méthodes avec $k > 1$, à chaque début de pas de temps de calcul n (même pour le premier calcul initial), nous avons besoin de connaître les valeurs du vecteur N_n mais aussi des valeurs précédentes N_{n-k}. Or, cela n'est pas possible au début de la simulation. Pour y remédier, avec ce type d'approche il faut penser à utiliser une procédure spécifique d'initialisation afin d'obtenir les valeurs demandées (comme commencer avec une autre méthode le temps d'avoir ces données). Cette méthode n'est donc pas *Free to (re)start*.

Expériences numériques pour GIF/RK

Il a été montré ([55] page 78) que l'erreur locale des méthodes *GIF/RK* décroît lorsque nous augmentons le degré du polynôme approximant la partie non linéaire et que nous gagnons plusieurs ordres de grandeur dans la précision des résultats par rapport à la méthode plus classique *IF RK*. Cela est vrai pour les simulations avec un pas de temps fixe, car dans le cas d'un pas de temps variable, nous aurions plutôt une augmentation de la taille du pas de temps pour une précision identique. Par contre, Krogstad [50] nous explique que cette amélioration se fait au détriment de la région de stabilité, ce qui se voit particulièrement bien avec l'équation de *KdV*, comme étudié dans le chapitre V.

Exemple 1 : GL1/RK4

Ce premier exemple est pour $k = 1$.

TABLE III.3 – Méthode *GL1/RK4*

0	0	0	0	0	$\mathbb{1}$
$\frac{1}{2}$	$\frac{1}{2}\varphi^{[1,2]}$	0	0	0	$\mathrm{e}^{\frac{1}{2}\mathbb{L}\Delta t}$
$\frac{1}{2}$	$\frac{1}{2}\varphi^{[1,3]} - \frac{1}{2}\mathbb{1}$	$\frac{1}{2}\mathbb{1}$	0	0	$\mathrm{e}^{\frac{1}{2}\mathbb{L}\Delta t}$
1	$\varphi^{[1,4]} - \varphi^{[0,2]}$	0	$\varphi^{[0,2]}$	0	$\mathrm{e}^{\mathbb{L}\Delta t}$
	$\varphi^{[1]} - \frac{2}{3}\varphi^{[0,2]} - \frac{1}{6}\mathbb{1}$	$\frac{1}{3}\varphi^{[0,2]}$	$\frac{1}{3}\varphi^{[0,2]}$	$\frac{1}{6}\mathbb{1}$	$\mathrm{e}^{\mathbb{L}\Delta t}$

Minchev [56] nous donne les informations suivantes :
- Pour $\mathbb{L} = 0$, cette méthode se réduit à une méthode classique de *RK4* ;
- Ces méthodes préservent les points fixes ;
- La seule différence avec la méthode traditionnelle de Lawson est la modification des termes de la première colonne de la matrice A et du premier élément de la matrice B.

Exemple 2 : GL2/RK4

Les performances de la méthode peuvent être améliorées en augmentant le degré d'approximation de N, ce qui se fait en augmentant le degré du polynôme c. Pour cela, il faut utiliser des approximations issues du passé, c'est-à-dire s'inspirer des méthodes à plusieurs pas de temps, ce qui est fait avec l'exemple suivant, pour $k = 2$. Ici, nous faisons passer d'une boucle de calcul à l'autre deux quantités, à savoir u_n et N_{n-1}.

TABLE III.4 – Méthode *GL2/RK4*

0	0	0	0	0	$\mathbb{1}$	0
$\frac{1}{2}$	$\frac{1}{2}\varphi^{[1,2]} + \frac{1}{4}\varphi^{[2,2]}$	0	0	0	$\mathrm{e}^{\frac{1}{2}\mathbb{L}\Delta t}$	$-\frac{1}{4}\varphi^{[2,2]}$
$\frac{1}{2}$	$\frac{1}{2}\varphi^{[1,3]} + \frac{1}{4}\varphi^{[2,3]} - \frac{3}{4}\mathbb{1}$	$\frac{1}{2}\mathbb{1}$	0	0	$\mathrm{e}^{\frac{1}{2}\mathbb{L}\Delta t}$	$-\frac{1}{4}\varphi^{[2,3]} + \frac{1}{4}\mathbb{1}$
1	$\varphi^{[1,4]} + \varphi^{[2,4]} - \frac{3}{2}\varphi^{[0,2]}$	0	$\varphi^{[0,2]}$	0	$\mathrm{e}^{\mathbb{L}\Delta t}$	$-\varphi^{[2,4]} + \frac{1}{2}\varphi^{[0,2]}$
	$\varphi^{[1]} + \varphi^{[2]} - \varphi^{[0,2]} - \frac{1}{3}\mathbb{1}$	$\frac{1}{3}\varphi^{[0,2]}$	$\frac{1}{3}\varphi^{[0,2]}$	$\frac{1}{6}\mathbb{1}$	$\mathrm{e}^{\mathbb{L}\Delta t}$	$-\varphi^{[2]} + \frac{1}{3}\varphi^{[0,2]} + \frac{1}{6}\mathbb{1}$
	$\mathbb{1}$	0	0	0	0	0

Encore une fois, la référence [56] nous donne les informations suivantes :
- Pour $\mathbb{L} = 0$, la méthode générale linéaire sous-jacente a été construite par Butcher [12] ;
- Cette méthode est la seule à étages d'ordre 2 qui permet de faire passer u_n et N_{n-1} de boucles de calculs en boucles de calculs ;
- Nous voyons que seule la première colonne de la matrice A et le premier élément de la première colonne de la matrice B diffèrent de la méthode de Lawson. Cette propriété est commune à toutes les méthodes *GL* qui font appel à une méthode *RK*.

4.3 Synthèse

Nous avons donc vu qu'il existe deux principales techniques pour résoudre numériquement le problème initial : les méthodes de Facteur Intégrant *IF* et les méthodes d'Exponential Time Differencing *ETD*. A cela s'ajoute la possibilité de lier ces changements de variables aux méthodes d'avance temporelle directement dans les coefficients de ces dernières, comme par exemple avec les méthodes *IF RK*. Ainsi, il est donné différentes techniques « clés en main » pour lesquelles il suffit d'implémenter le schéma temporel et le changement de variable donné.

Chapitre IV

Facteur Intégrant Modifié

Nous souhaitons améliorer le facteur intégrant classique étudié dans le chapitre précédent afin de réduire la raideur numérique présente lors des simulations d'équations non linéaires. Cela permettrait d'augmenter la taille des pas de temps de calcul et donc de diminuer le temps total d'une simulation. De plus, le lien fait entre le changement de variable et l'avance temporelle reste un inconvénient. Changer de type de méthodes de Runge-Kutta pour une raison quelconque, n'est possible qu'avec de nombreuses modifications, contrainte fortement restrictive. Il serait plus intéressant que l'utilisateur puisse alterner sans le moindre problème les différentes avances temporelles sans devoir trop modifier son code numérique. Pour cela, nous introduisons ici notre nouvelle approche ayant pour nom *facteur intégrant modifié*, notée *MIF*, et qui est explicitée en IV.1. Afin de nous permettre de calculer les dérivées nécessaires à notre méthode, nous faisons appel à l'approximation du *Dense Output* en IV.1.4. Les applications aux équations considérées dans ce manuscrit sont données dans la section IV.2 et nous expliquons notre comparaison des résultats de simulations entre les différentes méthodes à la section IV.3.

1 Présentation du facteur intégrant modifié

Partons de la définition du système précédent (III.15)

$$y'(t) + \mathbb{A}y(t) = \mathrm{N}(t, y(t))$$

avec $\mathbb{A}$ la matrice de la partie linéaire et N le terme non linéaire.
Mathématiquement, rien ne nous empêche d'ajouter une même quantité aux deux membres de l'équation. Cela revient simplement à ajouter « zéro ». Or, d'un point de vue numérique nous estimons que cette action peut avoir une incidence très importante. Soustrayons alors un certain polynôme P à l'équation d'évolution afin d'obtenir

$$y'(t) + \mathbb{A}y(t) - \mathrm{P}(t) = \mathrm{N}(t, y(t)) - \mathrm{P}(t) \tag{IV.1}$$

Comme pour le facteur intégrant classique, nous introduisons une nouvelle variable z, en posant

$$y\left(t\right)=\mathrm{e}^{\mathbb{A}\left(t_n-t\right)}z\left(t\right)+\int_{t_n}^{t}\mathrm{e}^{\mathbb{A}\left(t'-t\right)}\mathrm{P}\left(t'\right)\mathrm{d}t' \tag{IV.2}$$

Nous voyons que la relation entre les deux variables y et z est plus compliquée que celle du facteur intégrant classique (III.16), avec l'ajout de l'intégrale pour tenir compte du polynôme P, mais nous respectons toujours la condition initiale $z\left(t_n\right)=y\left(t_n\right)$ lors du début de chaque pas de calcul. Si nous dérivons cette relation, il nous vient l'équation de la dérivée qui peut se reécrire sous la forme

$$y'\left(t\right)=-\mathbb{A}\mathrm{e}^{\mathbb{A}\left(t_n-t\right)}z\left(t\right)+\mathrm{e}^{\mathbb{A}\left(t_n-t\right)}z'\left(t\right)-\int_{t_n}^{t}\mathbb{A}\mathrm{e}^{\mathbb{A}\left(t'-t\right)}\mathrm{P}\left(t'\right)\mathrm{d}t'+\mathrm{P}\left(t\right) \tag{IV.3}$$

soit

$$y'\left(t\right)+\mathbb{A}y\left(t\right)-\mathrm{P}\left(t\right)=\mathrm{e}^{\mathbb{A}\left(t_n-t\right)}z'\left(t\right) \tag{IV.4}$$

Maintenant, nous pouvons trouver l'équation d'évolution de z en fonction de y

$$\begin{aligned}z'\left(t\right)&=\mathrm{e}^{\mathbb{A}\left(t-t_n\right)}\left(y'\left(t\right)+\mathbb{A}y\left(t\right)-\mathrm{P}\left(t\right)\right)\\&=\mathrm{e}^{\mathbb{A}\left(t-t_n\right)}\left(\mathrm{N}\left(t,y\left(t\right)\right)-\mathrm{P}\left(t\right)\right)\end{aligned} \tag{IV.5}$$

La différence avec le facteur intégrant classique (III.19) est la présence de la fonction P dans le terme multiplié par l'exponentielle. Et c'est grâce à cela que nous espérons améliorer les performances des simulations en réduisant la raideur numérique.

Après avoir donné quelques explications en IV.1.1, nous verrons quel est le choix qui est fait pour ce polynôme en IV.1.2. Ensuite, nous donnerons la forme de la solution de l'équation d'évolution en IV.1.3 et la méthodologie pour obtenir certaines dérivées nécessaires en IV.1.4.

1.1 Explications du choix de la méthode

Un intégrateur temporel de type Runge-Kutta nous donne une solution qui est exacte pour un polynôme d'ordre inférieur ou égal à l'ordre de la méthode. Si nous souhaitons modifier la résolution temporelle afin de l'optimiser, avec l'ajout d'une fonction polynomiale comme nous le faisons, nous avons la possibilité de le faire soit après, soit avant le changement de variable de y en z. Dans le premier cas, nous avons l'équation d'évolution du facteur intégrant classique

$$z'\left(t\right)=\mathrm{e}^{\mathbb{A}\left(t-t_n\right)}\mathrm{N}\left(t,y\left(t\right)\right)$$

et en soustrayant le polynôme P après coup, nous obtiendrions quelque chose de la forme

$$z'\left(t\right)\propto\mathrm{e}^{\mathbb{A}\left(t-t_n\right)}\mathrm{N}\left(t,y\left(t\right)\right)-P\left(t\right)$$

En faisant ainsi, nous ne changeons rien puisque la solution est déjà exacte pour un polynôme d'ordre plus ou moins élevé selon le schéma numérique. Pour espérer avoir une incidence sur le calcul, il faudrait prendre en compte un polynôme P d'ordre beaucoup plus grand que l'ordre de la méthode de type Runge-Kutta utilisée, ce qui n'est ni intéressant en terme de temps de

calcul, ni en terme de portabilité. De ce fait, nous utilisons la seconde possibilité, qui est de prendre en compte notre modification avant le changement de variable, de manière à obtenir l'équation d'évolution

$$z'(t) = \mathrm{e}^{\mathbb{A}(t-t_n)} \left(\mathrm{N}(t, y(t)) - \mathrm{P}(t)\right)$$

Dans ce cas, nous ne pouvons pas dire à l'avance comment va se comporter la résolution numérique et nous supposons que cela sera bénéfique.

1.2 Choix du polynôme P

En ce qui concerne le polynôme P, pour le moment nous n'avons rien précisé à son sujet. Nous souhaitons en quelque sorte minimiser le membre de droite de l'équation d'évolution (IV.5). En effet, plus il sera petit, plus l'évolution temporelle sera aisée à simuler car nous aurons réduit la difficulté de calcul issue de ce terme. Par définition, nous savons que le terme non linéaire est

$$\mathrm{N}(t, y(t)) = y'(t) + \mathbb{A}y(t) \tag{IV.6}$$

La première idée est de tout simplement poser la définition suivante du polynôme P, afin d'annuler l'équation d'évolution (IV.5) au temps initial t_n d'une boucle de calcul $[t_n, t_{n+1}]$,

$$\mathrm{P}(t_n) = y'(t_n) + \mathbb{A}y(t_n) \tag{IV.7}$$

En procédant ainsi, nous voyons bien que la fonction $\mathrm{N}(t, y(t)) - P(t) = y'(t) + \mathbb{A}y(t) - \mathrm{P}(t)$ de l'équation d'évolution de z en (IV.5) s'annule en t_n, ce qui peut nous permettre de rendre les calculs moins difficiles à réaliser. Mais nous pouvons faire mieux. En effet, il est possible de faire un développement de Taylor de ce polynôme, autour du début de la boucle de calcul, au temps t_n :

$$\begin{aligned}
\mathrm{P}_q(t) &= \left[y'(t_n) + \mathbb{A}y(t_n)\right] + (t - t_n)\left[y''(t_n) + \mathbb{A}y'(t_n)\right] + \frac{(t-t_n)^2}{2!}\left[y'''(t_n) + \mathbb{A}y''(t_n)\right] \\
&\quad + \cdots + \frac{(t_q - t_n)^q}{q!}\left[y^{(q+1)}(t_n) + \mathbb{A}y^{(q)}(t_n)\right] \\
&= \mathrm{p}_0 + (t - t_n)\,\mathrm{p}_1 + \frac{(t-t_n)^2}{2!}\mathrm{p}_2 + \frac{(t-t_n)^3}{3!}\mathrm{p}_3 + \cdots + \frac{(t_q - t_n)^2}{q!}\mathrm{p}_q
\end{aligned} \tag{IV.8}$$

où P_q représente le développement de Taylor tronqué à l'ordre q et p_q représente le coefficient de l'ordre q.

Remarques

- Dans les faits, nous tronquerons ce développement aux ordres 0, 1 ou 2, notés P_0, P_1 et P_2.
- Nous notons le facteur intégrant modifié à l'ordre q du polynôme P : *MIFq*.
- Au lieu de prendre ce développement autour de t_n, nous pourrions prendre le temps intermédiaire entre t_n et t_{n+1}, soit en $t_n + \dfrac{\Delta t}{2}$, afin d'avoir une meilleure approximation des dérivées composant le polynôme P sur l'intervalle de temps $[t_n, t_{n+1}]$ d'une boucle de calcul. Par contre, cela ne serait possible qu'en faisant des concessions sur le temps de calcul par l'ajout de fonctions supplémentaires. Nous montrerons simplement qu'il est possible d'améliorer le facteur intégrant classique avec notre définition du polynôme, qui pourrait ne pas être la plus optimale.

1.3 Solution de l'équation d'évolution

La solution $y(t)$ en (IV.2) peut se reécrire à l'aide de l'intégrale suivante (développée ici à l'ordre $q = 3$ du polynôme P)

$$\begin{aligned}\int_{t_n}^{t} \mathrm{e}^{\mathbb{A}(t'-t)}\mathrm{P}(t')\,\mathrm{d}t' = &\left(1-\mathrm{e}^{\mathbb{A}(t_n-t)}\right)\frac{\mathrm{p}_0}{\mathbb{A}} + \left(\mathrm{e}^{\mathbb{A}(t_n-t)}-1-\mathbb{A}(t_n-t)\right)\frac{\mathrm{p}_1}{\mathbb{A}^2}\\ &+\left(-\mathrm{e}^{\mathbb{A}(t_n-t)}+1+\mathbb{A}(t_n-t)+\frac{\mathbb{A}^2(t_n-t)^2}{2!}\right)\frac{\mathrm{p}_2}{\mathbb{A}^3}\\ &+\left(\mathrm{e}^{\mathbb{A}(t_n-t)}-1-\mathbb{A}(t_n-t)-\frac{\mathbb{A}^2(t_n-t)^2}{2!}-\frac{\mathbb{A}^3(t_n-t)^3}{3!}\right)\frac{\mathrm{p}_3}{\mathbb{A}^4}\end{aligned} \tag{IV.9}$$

comme l'équation

$$y(t) = \mathrm{e}^{\mathbb{A}(t_n-t)}z(t) + \sum_{n=0}^{q}\left(-\mathrm{e}^{\mathbb{A}(t_n-t)}+\sum_{k=0}^{n}\frac{\mathbb{A}^k(t_n-t)^k}{k!}\right)(-1)^n\,\mathbb{A}^{-(n+1)}\mathrm{p}_n \tag{IV.10}$$

Nous vérifions aussi que pour le temps initial d'un pas de temps en $t = t_n$, nous obtenons bien $y(t_n) = z(t_n)$.

Lorsque nous voudrons passer de la variable z à la variable originelle y, si certaines valeurs des coefficients de la matrice des termes linéaires $\mathbb{A}$ seront nuls, nous serons face à une indétermination de calcul. Pour y remédier, nous *créons* les différentes fonctions suivantes, dites *Exponentielles Cardinales*,

$$\begin{aligned}\mathrm{ExpCard}_1 &= \begin{cases}\dfrac{\mathrm{e}^x-1}{x} & \text{si} \quad x\neq 0\\ 1 & \text{si} \quad x=0\end{cases}\\ \mathrm{ExpCard}_2 &= \begin{cases}\dfrac{\mathrm{e}^x-1-x}{x^2} & \text{si} \quad x\neq 0\\ \dfrac{1}{2} & \text{si} \quad x=0\end{cases}\\ \mathrm{ExpCard}_3 &= \begin{cases}\dfrac{\mathrm{e}^x-1-x-\frac{x^2}{2!}}{x^3} & \text{si} \quad x\neq 0\\ \dfrac{1}{6} & \text{si} \quad x=0\end{cases}\end{aligned} \tag{IV.11}$$

Cela nous permet de reécrire l'équation (IV.10) de y sous la forme

$$\begin{aligned}y(t) = \mathrm{e}^{\mathbb{A}(t_n-t)}z(t) &- \frac{\left[\mathrm{e}^{\mathbb{A}(t_n-t)}-1\right]}{\left[\mathbb{A}(t_n-t)\right]}(t_n-t)\,\mathrm{p}_0 + \frac{\left[\mathrm{e}^{\mathbb{A}(t_n-t)}-1-\mathbb{A}(t_n-t)\right]}{\left[\mathbb{A}(t_n-t)\right]^2}(t_n-t)^2\,\mathrm{p}_1\\ &- \frac{\left[\mathrm{e}^{\mathbb{A}(t_n-t)}-1-\mathbb{A}(t_n-t)-\mathbb{A}^2\frac{(t_n-t)^2}{2!}\right]}{\left[\mathbb{A}(t_n-t)\right]^3}(t_n-t)^3\,\mathrm{p}_2\\ &+ \frac{\left[\mathrm{e}^{\mathbb{A}(t_n-t)}-1-\mathbb{A}(t_n-t)-\mathbb{A}^2\frac{(t_n-t)^2}{2!}-\mathbb{A}^3\frac{(t_n-t)^3}{3!}\right]}{\left[\mathbb{A}(t_n-t)\right]^4}(t_n-t)^4\,\mathrm{p}_3+\cdots\end{aligned} \tag{IV.12}$$

1.4 Obtention des différentes dérivées : Dense Output

Le dernier point à expliciter est l'obtention des parties du polynôme P (IV.8) qui contiennent les dérivées $y'(t)$, $y''(t)$, $y'''(t)$ et $y''''(t)$. Comme nous ne pouvons pas les obtenir directement par la méthode d'avancement temporel, nous allons passer par une fonction intermédiaire qui approxime la solution et que nous dérivons.
En utilisant le *PI Step Control* vu en II.2.3, lors de l'exécution du code, l'ordinateur détermine la valeur optimale du pas de temps Δt. Cela veut dire que les temps dits de *sortie*, pour lesquels nous obtenons les valeurs de la solution simulée, ne peuvent pas être prédéfinis. Pour y remédier, nous allons utiliser une fonction dénommée *Dense Output* [41], qui nous permet de trouver une approximation par interpolation de la solution sur tout l'intervalle de temps $[t_n,\, t_{n+1}]$, ce qui nous permettra d'obtenir une approximation de la solution à des temps prédéfinis. Cela est fait de manière très peu coûteuse en terme de temps de calcul, voire négligeable lors de simulations fortement non linéaires, puisque nous ne faisons qu'assembler des données déjà calculées au préalable, les dérivées k_i. Le reste des opérations n'est que multiplications ou additions.

Définition de l'interpolation

Admettons qu'en partant de t_n, avec un pas de temps Δt qui nous amène en t_{n+1}, nous souhaitons obtenir la valeur de $y(t)$ en $t_\theta = t_n + \theta\Delta t$, définie telle que

$$t_n \leq t_\theta \leq t_{n+1}, \qquad 0 \leq \theta \leq 1 \tag{IV.13}$$

avec les conditions aux bords de l'interpolation temporelle

$$\begin{aligned} t_\theta|_{\theta=0} &= t_n \\ t_\theta|_{\theta=1} &= t_{n+1} \end{aligned} \tag{IV.14}$$

Nous avons $t_n + \Delta t = t_{n+1}$ et $t_\theta + \theta\Delta t = t_{n+1}$, donc il nous vient la définition suivante de θ :

$$\theta = \frac{t_\theta - t_n}{\Delta t} \tag{IV.15}$$

Maintenant que nous savons comment trouver le temps t_θ, il nous faut trouver la valeur de $y(t_\theta)$. Pour ce faire, à l'aide des dérivées k_i précédemment obtenues par la méthode de Runge-Kutta d'ordre p à s étages (voir II.2.2), nous interpolons cette valeur en posant la fonction $u(\theta)$

$$\begin{aligned} u(\theta) &= y_n + \Delta t \sum_{i=1}^{s*} b_i(\theta)\, k_i \\ u(0) &= y_n \\ u(1) &= y_{n+1} \end{aligned} \tag{IV.16}$$

et c'est grâce à cette fonction que nous pourrons obtenir des approximations de la solution à n'importe quel temps. Nous voyons que la somme en (IV.16) se fait sur $s^* \geq s+1$. Cela veut dire que, selon les cas, cette approximation peut nécessiter plus d'étapes d'évaluations intermédiaires k_i que pour la méthode de Runge-Kutta.

Calcul de dérivées

Dans le développement de notre intégrateur temporel, nous avons besoin de calculer des dérivées. Nous le faisons par le biais du Dense Output. En effet, il est possible de dériver $u(\theta)$ en (IV.16) par rapport à θ, en faisant attention au fait que $\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{1}{\Delta t}$, c'est-à-dire $\mathrm{d}\theta = \frac{\mathrm{d}t}{\Delta t}$ d'après (IV.15). Ainsi, nous obtenons les premières, secondes et troisièmes dérivées de $u(\theta)$

$$\begin{aligned}
\frac{\mathrm{d}u(\theta)}{\mathrm{d}t} &= \sum_{i=1}^{s*} \frac{\mathrm{d}b_i(\theta)}{\mathrm{d}\theta} k_i = f(t_\theta, y_\theta) = \left.\frac{\mathrm{d}y(t)}{\mathrm{d}t}\right|_{t_\theta} \\
\frac{\mathrm{d}^2u(\theta)}{\mathrm{d}t^2} &= \frac{1}{\Delta t}\sum_{i=1}^{s*} \frac{\mathrm{d}^2b_i(\theta)}{\mathrm{d}\theta^2} k_i = f^{(1)}(t_\theta, y_\theta) = \left.\frac{\mathrm{d}^2y(t)}{\mathrm{d}t^2}\right|_{t_\theta} \\
\frac{\mathrm{d}^3u(\theta)}{\mathrm{d}t^3} &= \frac{1}{\Delta t^2}\sum_{i=1}^{s*} \frac{\mathrm{d}^3b_i(\theta)}{\mathrm{d}\theta^3} k_i = f^{(2)}(t_\theta, y_\theta) = \left.\frac{\mathrm{d}^3y(t)}{\mathrm{d}t^3}\right|_{t_\theta}
\end{aligned} \tag{IV.17}$$

Il nous est donc possible de calculer les dérivées première $f^{(1)}$ et seconde $f^{(2)}$ de notre fonction $f(t,y) = y'(t)$ qui nous sont utiles.
Prenons deux boucles de temps consécutives, celle en $n-1$: $[t_{n-1}; t_n]$ et celle en n : $[t_n; t_{n+1}]$. Le temps t_n au pas de calcul $n-1$ étant égal au temps t_n du pas de calcul n, nous pouvons calculer ces dérivées au temps t_n du pas de calcul n, en utilisant les valeurs obtenues au temps t_n du pas de calcul $n-1$.

1.4.a *Dense Output de la méthode temporelle de Bogacki-Shampine*

Pour les méthodes de faible ordre, comme celle de Bogacki-Shampine, il existe un cas particulier de la formule du Dense Output (IV.16) qui fait appel à une interpolation cubique d'Hermite [41, 66]. Cette approximation de la solution y se note

$$u(\theta) = (1-\theta)y_0 + \theta y_1 + \theta(\theta-1)\left[(1-2\theta)(y_1-y_0) + (\theta-1)\Delta t f_0 + \theta\Delta t f_1\right] \tag{IV.18}$$

avec $f_0 = y'_0$ et $f_1 = y'_1$. A noter que nous avons toujours $u(0) = y_0$ et $u(1) = y_1$. Mais l'approximation ainsi obtenue ne nous permet d'obtenir que la dérivée première $u'(\theta)$, dont les cas limites sont $u'(0) = y'_0$ et $u'(1) = y'_1$. Il est certes possible de calculer la dérivée seconde, mais elle ne représente pas une approximation de y'' du fait d'erreurs numériques trop importantes.

1.4.b *Dense Output de la méthode temporelle de Dormand-Prince*

Coefficients du Dense Output

Puisque le schéma numérique utilisé possède $s = 6$ étages de calculs effectifs et que nous devons respecter la règle $s^* \geq s+1$, nous obtenons donc $s^* = 7$, d'où l'utilisation des coefficients k_1 à k_7 définis en (II.49). Donc cette approximation est *gratuite*, au sens où nous n'avons pas besoin de calculer de quantités k_i supplémentaires. Les coefficients $b_i(\theta)$ suivants nous sont

donnés par [41] :

$$\begin{aligned}
b_1(\theta) &= \theta^2(3-2\theta)b_1 + \theta(\theta-1)^2 - \theta^2(\theta-1)^2 5\,\frac{2558722523 - 31403016\,\theta}{11282082432}\\
b_2(\theta) &= 0\\
b_3(\theta) &= \theta^2(3-2\theta)b_3 + \theta^2(\theta-1)^2 100\,\frac{882725551 - 15701508\,\theta}{32700410799}\\
b_4(\theta) &= \theta^2(3-2\theta)b_4 - \theta^2(\theta-1)^2 25\,\frac{443332067 - 31403016\,\theta}{1880347072}\\
b_5(\theta) &= \theta^2(3-2\theta)b_5 + \theta^2(\theta-1)^2 32805\,\frac{23143187 - 3489224\,\theta}{199316789632}\\
b_6(\theta) &= \theta^2(3-2\theta)b_6 - \theta^2(\theta-1)^2 55\,\frac{29972135 - 7076736\,\theta}{822651844}\\
b_7(\theta) &= \theta^2(\theta-1) + \theta^2(\theta-1)^2 10\,\frac{7414447 - 829305\,\theta}{29380423}
\end{aligned} \tag{IV.19}$$

Cette méthode nous donne une approximation d'ordre 4 de la solution. Il serait possible d'atteindre l'ordre 5 avec l'ajout du calcul d'une fonction supplémentaire. Or, comme nous ne voulons pas perdre du temps de calcul, nous n'utiliserons pas cette ordre supérieur.

Dérivées de la solution approximée

Puisque ce Dense Output est d'ordre 4, cela veut dire que la première dérivée (IV.17) qui approxime $\left.\frac{\mathrm{d}y(t)}{\mathrm{d}t}\right|_{t_\theta}$ est d'ordre 3, la seconde qui approxime $\left.\frac{\mathrm{d}^2y(t)}{\mathrm{d}t^2}\right|_{t_\theta}$ est d'ordre 2 et la troisième dérivée qui approxime $\left.\frac{\mathrm{d}^3y(t)}{\mathrm{d}t^3}\right|_{t_\theta}$ est d'ordre 1. Nous voyons donc une limite apparaître, celle des dérivées utilisables. Ci-dessous nous donnons les valeurs des coefficients $b_i(\theta)$ en $\theta = 1$ qui nous serons utiles par la suite,

$$\frac{\mathrm{d}b_1}{\mathrm{d}\theta}(1) = 0, \quad \frac{\mathrm{d}b_3}{\mathrm{d}\theta}(1) = 0, \quad \frac{\mathrm{d}b_4}{\mathrm{d}\theta}(1) = 0, \quad \frac{\mathrm{d}b_5}{\mathrm{d}\theta}(1) = 0, \quad \frac{\mathrm{d}b_6}{\mathrm{d}\theta}(1) = 0, \quad \frac{\mathrm{d}b_7}{\mathrm{d}\theta}(1) = 1 \tag{IV.20}$$

$$\begin{aligned}
&\frac{\mathrm{d}^2b_1}{\mathrm{d}\theta^2}(1) = \frac{-2219729759}{2820520608}, \quad \frac{\mathrm{d}^2b_3}{\mathrm{d}\theta^2}(1) = \frac{85263539600}{32700410799}, \quad \frac{\mathrm{d}^2b_4}{\mathrm{d}\theta^2}(1) = \frac{-6985389575}{470086768}\\
&\frac{\mathrm{d}^2b_5}{\mathrm{d}\theta^2}(1) = \frac{418756605759}{49829197408}, \quad \frac{\mathrm{d}^2b_6}{\mathrm{d}\theta^2}(1) = \frac{-791215799}{205662961}, \quad \frac{\mathrm{d}^2b_7}{\mathrm{d}\theta^2}(1) = \frac{249224532}{29380423}
\end{aligned} \tag{IV.21}$$

$$\begin{aligned}
&\frac{\mathrm{d}^3b_1}{\mathrm{d}\theta^3}(1) = \frac{-248292612}{29380423}, \quad \frac{\mathrm{d}^3b_3}{\mathrm{d}\theta^3}(1) = \frac{284908469600}{10900136933}, \quad \frac{\mathrm{d}^3b_4}{\mathrm{d}\theta^3}(1) = \frac{-4173702325}{58760846}\\
&\frac{\mathrm{d}^3b_5}{\mathrm{d}\theta^3}(1) = \frac{122207108373}{3114324838}, \quad \frac{\mathrm{d}^3b_6}{\mathrm{d}\theta^3}(1) = \frac{-3517094768}{205662961}, \quad \frac{\mathrm{d}^3b_7}{\mathrm{d}\theta^3}(1) = \frac{916741278}{29380423}
\end{aligned} \tag{IV.22}$$

1.4.c *Dense Output de la méthode temporelle de Verner*

Coefficients du Dense Output

Avec ce type de méthode d'avance temporelle de haut degré, nous avons la même relation que (IV.16), mais il existe une étape supplémentaire, les coefficients $b_i(\theta)$ étant définis par [74, 77]

$$b_i(\theta) = \sum_{r=1}^{p^*} b_{i,r}\theta^r \tag{IV.23}$$

où p^* est l'ordre de la méthode d'interpolation. Ici ce sera $p^* = 8$, tout en sachant qu'il existe une méthode d'ordre 9 mais qui n'est pas satisfaisante sur la précision à long terme [74]. Nous devons faire $s^* = 21$ étages de calculs (pour l'ordre 8 contre 26 pour l'ordre 9), soit 5 de plus par rapport à la méthode de Verner elle-même. Ce coût de calcul supplémentaire est non négligeable, mais nous utiliserons tout de même cette technique pour tester notre facteur intégrant modifié avec cette avance temporelle d'ordre élevé.

Dérivées de la solution approximée

Les dérivées sont identiques au cas précédent (IV.17), avec l'ajout des étapes de calcul supplémentaires, comme nous le voyons avec les relations

$$\begin{aligned}
\frac{\mathrm{d}b_i(\theta)}{\mathrm{d}\theta} &= \frac{\mathrm{d}}{\mathrm{d}\theta}\left(\sum_{r=1}^{p^*} b_{i,r}\theta^r\right) = \sum_{r=1}^{p^*} b_{i,r}\frac{\mathrm{d}}{\mathrm{d}\theta}(\theta^r) = \sum_{r=1}^{p^*} b_{i,r}r\theta^{r-1} \\
\frac{\mathrm{d}^2b_i(\theta)}{\mathrm{d}\theta^2} &= \frac{\mathrm{d}^2}{\mathrm{d}\theta^2}\left(\sum_{r=1}^{p^*} b_{i,r}\theta^r\right) = \sum_{r=1}^{p^*} b_{i,r}\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}(\theta^r) = \sum_{r=1}^{p^*} b_{i,r}r(r-1)\theta^{r-2} \\
\frac{\mathrm{d}^3b_i(\theta)}{\mathrm{d}\theta^3} &= \frac{\mathrm{d}^3}{\mathrm{d}\theta^3}\left(\sum_{r=1}^{p^*} b_{i,r}\theta^r\right) = \sum_{r=1}^{p^*} b_{i,r}\frac{\mathrm{d}^3}{\mathrm{d}\theta^3}(\theta^r) = \sum_{r=1}^{p^*} b_{i,r}r(r-1)(r-2)\theta^{r-3}
\end{aligned} \tag{IV.24}$$

Au final nous obtenons les dérivées

$$\begin{aligned}
\frac{\mathrm{d}u(\theta)}{\mathrm{d}t} &= \sum_{i=1}^{s*}\frac{\mathrm{d}b_i(\theta)}{\mathrm{d}\theta}k_i = \sum_{i=1}^{s*}\left(\sum_{r=1}^{p^*} b_{i,r}r\theta^{r-1}\right)k_i \\
\frac{\mathrm{d}^2u(\theta)}{\mathrm{d}t^2} &= \frac{1}{\Delta t}\sum_{i=1}^{s*}\frac{\mathrm{d}^2b_i(\theta)}{\mathrm{d}\theta^2}k_i = \frac{1}{\Delta t}\sum_{i=1}^{s*}\left(\sum_{r=1}^{p^*} b_{i,r}r(r-1)\theta^{r-2}\right)k_i \\
\frac{\mathrm{d}^3u(\theta)}{\mathrm{d}t^3} &= \frac{1}{\Delta t^2}\sum_{i=1}^{s*}\frac{\mathrm{d}^3b_i(\theta)}{\mathrm{d}\theta^3}k_i = \frac{1}{\Delta t^2}\sum_{i=1}^{s*}\left(\sum_{r=1}^{p^*} b_{i,r}r(r-1)(r-2)\theta^{r-3}\right)k_i
\end{aligned} \tag{IV.25}$$

Puisque ce Dense Output est d'ordre 8, cela veut dire que la première dérivée qui approxime $\left.\frac{\mathrm{d}y(t)}{\mathrm{d}t}\right|_{t_\theta}$ est d'ordre 7, la seconde qui approxime $\left.\frac{\mathrm{d}^2y(t)}{\mathrm{d}t^2}\right|_{t_\theta}$ est d'ordre 6 et la troisième dérivée qui approxime $\left.\frac{\mathrm{d}^3y(t)}{\mathrm{d}t^3}\right|_{t_\theta}$ est d'ordre 5.

1.4.d *Application du Dense Output*

Première dérivée

Si nous regardons de plus près la définition du polynôme, nous voyons que pour l'ordre 0 (uniquement avec le terme p_0), nous n'avons pas besoin de calculer une nouvelle quantité, puisque le terme $y'(t_n)+\mathbb{A}y(t_n)$ au pas de temps n nous est donné par l'intégrateur temporel du Runge-Kutta au dernier temps du pas de calcul $n-1$. Donc pour l'obtention de cette première dérivée, nous utilisons uniquement des quantités déjà calculées. Cela veut dire que le facteur intégrant modifié à l'ordre 0 est *gratuit* en terme de calculs numériques.

Seconde dérivée

Pour le polynôme à l'ordre 1 (avec les termes p_0 et p_1), nous repartons de (IV.4) qui se met sous la forme

$$y'(t)+\mathbb{A}y(t)=\mathrm{e}^{\mathbb{A}(t_n-t)}z'+\mathrm{P} \tag{IV.26}$$

En dérivant cette relation, nous obtenons la dérivée seconde de y

$$y''(t)+\mathbb{A}y'(t)=-\mathbb{A}\mathrm{e}^{\mathbb{A}(t_n-t)}z'+\mathrm{e}^{\mathbb{A}(t_n-t)}z''+\mathrm{P}' \tag{IV.27}$$

avec $\mathrm{P}'(t)=\mathrm{p}_1+(t-t_n)\,\mathrm{p}_2+\cdots$.
$z'(t)$ étant égale à la dernière dérivée calculée par la méthode de Runge-Kutta, soit par exemple k_7 d'après (II.49) pour l'avance temporelle de Dormand et Prince, nous utiliserons cette donnée déjà calculée en posant $z'(t)=k_7$. Enfin, $z''(t)$ nous est donnée par l'approximation du *Dense Output*. Puisqu'en utilisant le schéma numérique de Bogacki et Shampine, il n'est pas possible d'obtenir cette approximation de dérivée, nous ne pouvons donc étudier que le facteur intégrant modifié à l'ordre 0 avec cette avance temporelle.

Troisième dérivée

Enfin, pour le polynôme à l'ordre 2 (avec les termes p_0, p_1 et p_2), nous faisons de même pour trouver la troisième dérivée

$$y'''(t)+\mathbb{A}y''(t)=\mathbb{A}^2\mathrm{e}^{\mathbb{A}(t_n-t)}z'-2\mathbb{A}\mathrm{e}^{\mathbb{A}(t_n-t)}z''+\mathrm{e}^{\mathbb{A}(t_n-t)}z'''+\mathrm{P}'' \tag{IV.28}$$

avec $\mathrm{P}''(t)=\mathrm{p}_2+\cdots$ et $z'''(t)$ qui est une dérivée donnée par l'approximation du *Dense Output*.

Remarques

Nous pourrions très bien continuer ainsi de suite si nous n'avions pas une limite issue du *Dense Output*. L'approximation donnée par ce dernier étant un polynôme d'ordre fini, nous ne pourrons pas utiliser une infinité de dérivées. En pratique, nous verrons que nous devrons arrêter le développement du polynôme P à l'ordre 2 à cause d'erreurs numériques. De plus, en imposant un polynôme de fort degré, nous nous éloignerions trop de la solution exacte dans les équations étudiées, ce qui impliquerait un ajout de raideur numérique à la simulation, alors que nous voulons obtenir le contraire en développant cette nouvelle méthode.

Nous calculons ces dérivées au temps final t_{n+1} du pas de calcul n, pour nous en servir dans le pas de calcul suivant au temps initial t_{n+1}.

Au tout premier calcul initial de la simulation, en $t = t_{initial}$, nous ne pouvons pas procéder de cette manière par absence de données issues du pas de temps précédent. Nous posons alors $\mathrm{P}\left(t_{initial}\right) = 0$ pour attendre d'avoir des données issues du passé après la première boucle de calculs, afin de pouvoir utiliser notre facteur intégrant modifié dès la boucle de calculs suivante. Cela ne pose pas de problèmes puisque nous commençons nos simulations numériques avec un pas de temps très petit, ce qui fait que cette attente sera totalement négligeable.

1.5 Liens avec le facteur intégrant généralisé

En créant notre facteur intégrant modifié, nous retrouvons l'idée de la méthode de facteur intégrant généralisé de Krogstad [50] décrite à la section III.4.2.b. Nous rappelons que l'équivalent de notre polynôme P dans la méthode de Krogstad, le polynôme c (voir (III.50)), prend les formes suivantes aux ordres 0, 1 et 2

$$
\begin{aligned}
c_0(\tau) &= N_n \\
c_1(\tau) &= N_n + \tau \left(\frac{N_n - N_{n-1}}{\Delta t_n} \right) \\
c_2(\tau) &= N_n + \tau \left(\frac{3N_n - 3N_{n-1}}{2\Delta t_n} - \frac{N_{n-1} - N_{n-2}}{2\Delta t_{n-1}} \right) + \tau^2 \left(\frac{N_n - N_{n-1}}{\Delta t_n^2} - \frac{N_{n-1} - N_{n-2}}{\Delta t_n \Delta t_{n-1}} \right)
\end{aligned}
$$

avec $\tau = t - t_n$. En ce qui concerne l'ordre 2, nous avons modifié l'approche initiale pour tenir compte d'un pas de temps adaptatif (voir (III.53)), avec deux pas de temps successifs différents Δt_{n-1} et Δt_n.

Point commun et différences entre les deux méthodes

- Le facteur intégrant modifié et le facteur intégrant généralisé sont identiques à l'ordre 0.
- Pour les ordres supérieurs, nous *devrions* avoir un meilleur calcul des dérivées à l'aide de l'approximation du Dense Output sur la boucle de temps actuel, par rapport à des différences finies sur les pas de temps précédents pour le facteur intégrant généralisé.
- Nous n'avons pas besoin de stocker des matrices issues de plusieurs pas de calcul précédents car nous ne faisons appel qu'à des quantités calculées au pas de temps courant et les nouvelles quantités obtenues n'étant pas à garder, nous pouvons les effacer une fois le calcul du polynôme effectué.
- Notre méthode est donc *one-step* à tous les ordres alors que celle du facteur intégrant modifié est *multistep* dès l'ordre 1, puisqu'elle fait appel à des données issues de deux temps successifs.
- Nous avons une méthode totalement indépendante de l'avance temporelle de Runge-Kutta choisie. Il est même tout à fait possible d'utiliser le facteur intégrant modifié avec un autre type de méthodes, comme, par exemple, celles de *Rosenbrock*. Par contre, la méthode du facteur intégrant généralisé est décrite uniquement pour une approche de Runge-Kutta d'ordre 4 (voir les tableaux III.3 et III.4). Dans cette thèse, nous montrons que l'indépendance entre facteur intégrant modifié et méthode temporelle est un aspect très important.

Utilisation du facteur intégrant généralisé

Comme nous venons de le dire, le facteur intégrant généralisé a été développé dans le but de le lier à *la* méthode de Runge-Kutta d'ordre 4 (voir (II.45)), qui est la plus célèbre approche de ce type. Et ce, afin d'être une alternative aux autres méthodes d'intégrateurs exponentiels d'ordre 4, dont la plus connue est la méthode *ETDRK4* (voir III.1.2 et [23]). Or, il s'avère que ce n'est pas nécessairement la plus efficace des méthodes de Runge-Kutta. En effet, selon les cas étudiés, il est nécessaire d'utiliser une méthode différente. Par exemple, soit pour augmenter ou diminuer l'ordre de la résolution, soit pour utiliser des schémas emboîtés afin de faire appel à un pas de temps adaptatif efficace (voir II.2.3). Comme nous n'utilisons pas la méthode de Runge-Kutta d'ordre 4, nous avons adapté le facteur intégrant généralisé aux autres méthodes décrites en II.2.2. Ainsi, pour toutes les simulations que nous réalisons avec notre facteur intégrant modifié aux ordres 1 et 2, nous effectuons aussi des tests avec le facteur intégrant généralisé, en remplaçant le calcul des dérivées en IV.1.4.d par les différences finies des relations (III.50) et (III.53). Cela nous permet de voir s'il existe des différences dans les résultats, vis-à-vis de notre facteur intégrant modifié.

Nous avons expliqué en IV.1.4.a qu'avec l'approche Runge-Kutta de Bogacki-Shampine, le Dense Output ne nous permet pas d'obtenir les dérivées nécessaires aux ordres 1 et 2 de notre facteur intégrant modifié. Donc nous ne pourrons l'utiliser qu'à l'ordre 0. Par contre, puisque le calcul des dérivées est fait d'une autre manière pour le facteur intégrant généralisé, il est possible de l'utiliser à ces ordres, même avec ce schéma de Runge-Kutta, du moment que nous avons à notre disposition les quantités N_n, N_{n-1} et N_{n-2}. De ce fait, afin de se donner une idée de l'efficacité d'un facteur intégrant modifié aux ordres autres que 0, nous pouvons regarder les performances du facteur intégrant généralisé puisque nos approches respectives se ressemblent.

2 Applications du facteur intégrant modifié

Pour nos simulations numériques, nous allons donc utiliser la mise sous forme de facteur intégrant modifié des équations des vagues considérées. Après avoir décrit la procédure temporelle globale en IV.2.1 et pour mieux visualiser ce que nous faisons, nous allons donner en exemple les termes issus des équations qui seront traitées par la suite. Dans la sous-section IV.2.2 nous donnons les applications du facteur intégrant modifié pour les équations de *KdV*, *BBM* et *NLS* et dans la sous-section IV.2.3 nous faisons de même pour les équations couplées de *Serre* et du modèle *HOS*.

2.1 Procédure pour l'intégration temporelle

L'intégration temporelle est implémentée de la manière suivante :
• Nous séparons les parties linéaire et non linéaire de l'équation d'évolution considérée.
• Nous appliquons un changement de variable, le facteur intégrant modifié, afin d'intégrer analytiquement la partie linéaire du système.
• Nous calculons les termes non linéaires dans l'espace physique et utilisons un anti-aliasing à chaque calcul pour prévenir les erreurs numériques (voir II.1.3).
• Nous utilisons une méthode de Runge-Kutta à pas de temps adaptatif pour faire évoluer les parties non linéaires restantes (voir II.2.2).

• A la fin de chaque boucle de calculs nous inversons le changement de variable pour revenir dans la variable originelle.

2.2 Equations de *KdV*, *BBM* et *NLS*

Les détails de ces équations sont disponibles dans le chapitre I.2 pour l'espace physique et en II.1.4 pour l'espace de Fourier. Nous les rappelons dans les sous-sections IV.2.2.a, IV.2.2.b et IV.2.2.c. Dans les sous-sections IV.2.2.d et IV.2.2.e nous donnons de manière générique les relations entre les variables y et z, ainsi que celles menant aux calculs des dérivées nécessaires pour notre méthode, en posant le vecteur du terme linéaire $\mathbb{A} = \mathrm{i}\ \omega$, avec ω un vecteur de nombres réels et $\tau = t - t_n$.

2.2.a *Equation de Korteweg et de Vries*

Il nous est possible d'écrire l'équation de KdV définie en II.1.4.a en séparant parties linéaire et non linéaire, comme

$$\widehat{\eta}_t + \mathbb{A}\widehat{\eta} = -\mathrm{i}k\frac{3}{4}\sqrt{\frac{g}{d}}\mathcal{F}\left\{\eta^2\right\} \tag{IV.29}$$

avec le vecteur du terme linéaire défini par $\mathbb{A} = c_0\mathrm{i}k\left(1 - k^2\frac{d^2}{6}\right)$.

2.2.b *Equation de Benjamin Bona et Mahony*

En posant le vecteur $\mathbb{A} = \mathrm{i}\dfrac{kc_0}{1 + k^2\frac{d^2}{6}}$, nous pouvons reécrire l'équation (II.10) de la section II.1.4.b en séparant parties linéaire et non linéaire comme

$$\widehat{\eta}_t + \mathbb{A}\widehat{\eta} = -\mathrm{i}k\frac{\frac{3}{4}\sqrt{\frac{g}{d}}}{1 + k^2\frac{d^2}{6}}\mathcal{F}\left\{\eta^2\right\} \tag{IV.30}$$

2.2.c *Equation de Schrödinger Non Linéaire*

Nous pouvons reécrire l'équation de NLS, vue en II.1.4.c, en séparant parties linéaire et non linéaire dans le repère fixe

$$\widehat{\psi}_t + \mathbb{A}\widehat{\psi} = -\frac{\mathrm{i}}{2}\omega_0 k_0^2\mathcal{F}\left\{|\psi|^2\psi\right\} \tag{IV.31}$$

avec la relation de dispersion $\omega = c_g k\left(1 - \dfrac{k}{4k_0}\right)$ et faire de même pour le repère mobile

$$\widehat{\psi}_t + \mathbb{A}\widehat{\psi} = -\frac{\mathrm{i}}{2}\omega_0 k_0^2\mathcal{F}\left\{|\psi|^2\psi\right\} \tag{IV.32}$$

avec la relation de dispersion $\omega = -c_g\dfrac{k^2}{4k_0}$.

2.2.d *Relation entre les variables y et z*

Nous pouvons reécrire la relation entre les variables y et z (IV.12) comme ce qui suit, en coupant le développement polynomial à l'ordre 2,

$$\begin{aligned} y(t) &= (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z(t) - \frac{[\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau) - 1]}{\mathrm{i}\omega}\, \mathrm{P}_0 \\ &+ \frac{[\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau) - 1 + \mathrm{i}\ \omega\tau]}{-\omega^2}\, \mathrm{P}_1 - \frac{\left[\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau) - 1 + \mathrm{i}\ \omega\tau + \omega^2 \frac{\tau^2}{2}\right]}{-\mathrm{i}\omega^3}\, \mathrm{P}_2 \\ &= (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z(t) - \frac{\mathrm{P}_0}{\omega}\left[\sin(\omega\tau) + \mathrm{i}\ (\cos(\omega\tau) - 1)\right] \\ &+ \frac{\mathrm{P}_1}{\omega^2}\left[-\cos(\omega\tau) + 1 + \mathrm{i}\ (\sin(\omega\tau) - \omega\tau)\right] \\ &- \frac{\mathrm{P}_2}{\omega^3}\left[\sin(\omega\tau) - \omega\tau + \mathrm{i}\ (\cos(\omega\tau) - 1 + \omega^2\frac{\tau^2}{2})\right] \end{aligned} \tag{IV.33}$$

et pour le cas où des composantes de $\mathbb{A}$ seraient nulles, à l'aide des exponentielles cardinales (IV.11) nous obtenons

$$y(t) = z(t) + \tau\ \mathrm{p}_0 + \frac{\tau^2}{2}\ \mathrm{p}_1 + \frac{\tau^3}{6}\ \mathrm{p}_2 \tag{IV.34}$$

2.2.e *Calcul des dérivées de y*

Nous pouvons aussi obtenir les dérivées (IV.26) à (IV.28), qui se reécrivent comme

$$\begin{aligned} y'(t) + \mathbb{A}y(t) &= (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z' + \mathrm{P} \\ y''(t) + \mathbb{A}y'(t) &= -\mathrm{i}\ \omega\, (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z' + (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z'' + \mathrm{P}' \\ y'''(t) + \mathbb{A}y''(t) &= -\omega^2\, (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z' \\ &\quad - 2\mathrm{i}\ \omega\, (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z'' + (\cos(\omega\tau) - \mathrm{i}\ \sin(\omega\tau))\, z''' + \mathrm{P}'' \end{aligned} \tag{IV.35}$$

Ce sont celles que nous utiliserons dans le calcul de notre polynôme P aux différents ordres.

2.3 Equations de *Serre* et du modèle *High-Order Spectral*

Les détails de ces équations sont disponibles dans le chapitre I.2 pour l'espace physique et en II.1.4 pour l'espace de Fourier. Nous les rappelons dans les sous-sections IV.2.3.b et IV.2.3.c. Dans la sous-section IV.2.3.a nous donnons des éléments de résolution communs à ces deux modèles.

2.3.a *Présentation générale*

Considérons le cas de deux équations différentielles couplées dont les deux variables sont notées y_1 et y_2. Prenons une matrice $\mathbb{A} = \begin{pmatrix} 0 & \mathbb{A}_{12} \\ \mathbb{A}_{21} & 0 \end{pmatrix}$ anti-diagonale, car nous n'aurons pas de

termes $\mathbb{A}_{11}$ et $\mathbb{A}_{22}$. Le système des équations d'évolutions pour le cas de deux équations couplées peut se mettre sous la forme

$$\begin{pmatrix} y_a^{'} \\ y_b^{'} \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_a \\ y_b \end{pmatrix} = \begin{pmatrix} \mathrm{N}_a\left(y\left(t\right)\right) \\ \mathrm{N}_b\left(y\left(t\right)\right) \end{pmatrix} \tag{IV.36}$$

et l'expression du polynôme P à l'ordre 2 devient

$$\begin{aligned} \begin{pmatrix} \mathrm{P}_a\left(t\right) \\ \mathrm{P}_b\left(t\right) \end{pmatrix} &= \left[\begin{pmatrix} y_a' \\ y_b' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_a \\ y_b \end{pmatrix} \right] + \left(t - t_n\right) \left[\begin{pmatrix} y_a'' \\ y_b'' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_a' \\ y_b' \end{pmatrix} \right] \\ &\quad + \frac{\left(t - t_n\right)^2}{2} \left[\begin{pmatrix} y_a''' \\ y_b''' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_a'' \\ y_b'' \end{pmatrix} \right] \\ &= \begin{pmatrix} \mathrm{P0}_a \\ \mathrm{P0}_b \end{pmatrix} + \left(t - t_n\right) \begin{pmatrix} \mathrm{P1}_a \\ \mathrm{P1}_b \end{pmatrix} + \frac{\left(t - t_n\right)^2}{2} \begin{pmatrix} \mathrm{P2}_a \\ \mathrm{P2}_b \end{pmatrix} \end{aligned} \tag{IV.37}$$

La relation (IV.12) du changement de variable de y en z s'écrit alors comme

$$\begin{aligned} \begin{pmatrix} y_a \\ y_b \end{pmatrix} &= \mathrm{e}^{\mathbb{A}\left(t_n - t\right)} \begin{pmatrix} z_a \\ z_b \end{pmatrix} \\ &\quad - \frac{\left[\mathrm{e}^{\mathbb{A}\left(t_n - t\right)} - \mathbb{1}\right]}{\left[\mathbb{A}\left(t_n - t\right)\right]} \left(t_n - t\right) \begin{pmatrix} \mathrm{P0}_a \\ \mathrm{P0}_b \end{pmatrix} \\ &\quad + \frac{\left[\mathrm{e}^{\mathbb{A}\left(t_n - t\right)} - \mathbb{1} - \mathbb{A}\left(t_n - t\right)\right]}{\left[\mathbb{A}\left(t_n - t\right)\right]^2} \left(t_n - t\right)^2 \begin{pmatrix} \mathrm{P1}_a \\ \mathrm{P1}_b \end{pmatrix} \\ &\quad - \frac{\left[\mathrm{e}^{\mathbb{A}\left(t_n - t\right)} - \mathbb{1} - \mathbb{A}\left(t_n - t\right) - \mathbb{A}^2 \frac{\left(t_n - t\right)^2}{2}\right]}{\left[\mathbb{A}\left(t_n - t\right)\right]^3} \left(t_n - t\right)^3 \begin{pmatrix} \mathrm{P2}_a \\ \mathrm{P2}_b \end{pmatrix} \end{aligned} \tag{IV.38}$$

avec la matrice identité $\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Enfin, le système des équations d'évolution en z se note

$$\begin{pmatrix} z_a^{'} \\ z_b^{'} \end{pmatrix} = \mathrm{e}^{\mathbb{A}\left(t - t_n\right)} \begin{pmatrix} \mathrm{N}_a\left(y\left(t\right)\right) - \mathrm{P}_a\left(t\right) \\ \mathrm{N}_b\left(y\left(t\right)\right) - \mathrm{P}_b\left(t\right) \end{pmatrix} \tag{IV.39}$$

et nous le résoudrons numériquement à l'aide d'une avance temporelle de type Runge-Kutta.

2.3.b *Equations de Serre*

D'après les deux équations définies en II.1.4.d, nous pouvons reécrire le système sous forme matricielle en séparant parties linéaire et non linéaire

$$\partial_t \begin{pmatrix} \mathrm{i}k\widehat{\eta} \\ \dfrac{\mathrm{i}\omega}{g}\widehat{q} \end{pmatrix} + \mathbb{A} \begin{pmatrix} \mathrm{i}k\widehat{\eta} \\ \dfrac{\mathrm{i}\omega}{g}\widehat{q} \end{pmatrix} = \begin{pmatrix} k^2\ \mathcal{F}\left\{\eta u\right\} + \dfrac{k^2 d}{1 + \frac{k^2 d^2}{3}} \mathcal{F}\{h\eta_x u_x + \dfrac{1}{3}\left(\eta^2 + 2\eta d\right) u_{xx}\} \\ -\dfrac{\omega k}{g}\ \mathcal{F}\left\{\dfrac{1}{2}u^2 - gh + \dfrac{1}{2}h^2 u_x^2 - uq\right\} \end{pmatrix} \tag{IV.40}$$

avec $\omega = \sqrt{\dfrac{gk^2 d}{1 + \frac{k^2 d^2}{3}}}$, la matrice des parties linéaires $\mathbb{A} = \begin{pmatrix} 0 & \mathrm{i}\omega \\ \mathrm{i}\omega & 0 \end{pmatrix}$, ses puissances

$$\mathbb{A}^2 = \begin{pmatrix} -\omega^2 & 0 \\ 0 & -\omega^2 \end{pmatrix},\ \mathbb{A}^{-1} = \begin{pmatrix} 0 & -\dfrac{\mathrm{i}}{\omega} \\ -\dfrac{\mathrm{i}}{\omega} & 0 \end{pmatrix},\ \mathbb{A}^{-3} = \begin{pmatrix} 0 & \dfrac{\mathrm{i}}{\omega^3} \\ \dfrac{\mathrm{i}}{\omega^3} & 0 \end{pmatrix} \tag{IV.41}$$

et les exponentielles

$$\mathrm{e}^{\mathbb{A}t} = \begin{pmatrix} \cos(\omega\ t) & \mathrm{i}\ \sin(\omega\ t) \\ \mathrm{i}\ \sin(\omega\ t) & \cos(\omega\ t) \end{pmatrix}, \qquad \mathrm{e}^{-\mathbb{A}t} = \begin{pmatrix} \cos(\omega\ t) & -\mathrm{i}\ \sin(\omega\ t) \\ -\mathrm{i}\ \sin(\omega\ t) & \cos(\omega\ t) \end{pmatrix} \tag{IV.42}$$

Relation entre les variables y et z

Comme pour les équations précédentes, nous pouvons obtenir la relation entre y et z dans le cas général

$$\begin{aligned}
\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\ \sin(\omega\tau) \\ -\mathrm{i}\ \sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\
&\quad - \begin{pmatrix} \cos(\omega\tau) - 1 & -\mathrm{i}\ \sin(\omega\tau) \\ -\mathrm{i}\ \sin(\omega\tau) & \cos(\omega\tau) - 1 \end{pmatrix} \begin{pmatrix} 0 & -\dfrac{\mathrm{i}}{\omega} \\ -\dfrac{\mathrm{i}}{\omega} & 0 \end{pmatrix} \begin{pmatrix} p_{0a} \\ p_{0b} \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) - 1 & \mathrm{i}\ (-\sin(\omega\tau) + \omega\tau) \\ \mathrm{i}\ (-\sin(\omega\tau) + \omega\tau) & \cos(\omega\tau) - 1 \end{pmatrix} \begin{pmatrix} -\dfrac{1}{\omega^2} & 0 \\ 0 & -\dfrac{1}{\omega^2} \end{pmatrix} \begin{pmatrix} p_{1a} \\ p_{1b} \end{pmatrix} \\
&\quad - \begin{pmatrix} \cos(\omega\tau) - 1 + \dfrac{1}{2}\omega^2\tau^2 & \mathrm{i}\ (-\sin(\omega\tau) + \omega\tau) \\ \mathrm{i}\ (-\sin(\omega\tau) + \omega\tau) & \cos(\omega\tau) - 1 + \dfrac{1}{2}\omega^2\tau^2 \end{pmatrix} \begin{pmatrix} 0 & \dfrac{\mathrm{i}}{\omega^3} \\ \dfrac{\mathrm{i}}{\omega^3} & 0 \end{pmatrix} \begin{pmatrix} p_{2a} \\ p_{2b} \end{pmatrix} \\
&= \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\ \sin(\omega\tau) \\ -\mathrm{i}\ \sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\
&\quad - \frac{1}{\omega} \begin{pmatrix} -\sin(\omega\tau) & -\mathrm{i}\ (\cos(\omega\tau) - 1) \\ -\mathrm{i}\ (\cos(\omega\tau) - 1) & -\sin(\omega\tau) \end{pmatrix} \begin{pmatrix} p_{0a} \\ p_{0b} \end{pmatrix} \\
&\quad + \frac{1}{\omega^2} \begin{pmatrix} -\cos(\omega\tau) + 1 & \mathrm{i}\ (\sin(\omega\tau) - \omega\tau) \\ \mathrm{i}\ (\sin(\omega\tau) - \omega\tau) & -\cos(\omega\tau) + 1 \end{pmatrix} \begin{pmatrix} p_{1a} \\ p_{1b} \end{pmatrix} \\
&\quad - \frac{1}{\omega^3} \begin{pmatrix} \sin(\omega\tau) - \omega\tau & \mathrm{i}\ (\cos(\omega\tau) - 1 + \dfrac{1}{2}\omega^2\tau^2) \\ \mathrm{i}\ (\cos(\omega\tau) - 1 + \dfrac{1}{2}\omega^2\tau^2) & \sin(\omega\tau) - \omega\tau \end{pmatrix} \begin{pmatrix} p_{2a} \\ p_{2b} \end{pmatrix}
\end{aligned} \tag{IV.43}$$

ainsi que pour le cas où des composantes de $\mathbb{A}$ seraient nulles, à l'aide des exponentielles cardinales (IV.11) nous obtenons

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \begin{pmatrix} -\tau & 0 \\ 0 & -\tau \end{pmatrix} \begin{pmatrix} p_{0a} \\ p_{0b} \end{pmatrix} + \begin{pmatrix} \dfrac{\tau^2}{2} & 0 \\ 0 & \dfrac{\tau^2}{2} \end{pmatrix} \begin{pmatrix} p_{1a} \\ p_{1b} \end{pmatrix} - \begin{pmatrix} -\dfrac{\tau^3}{6} & 0 \\ 0 & -\dfrac{\tau^3}{6} \end{pmatrix} \begin{pmatrix} p_{2a} \\ p_{2b} \end{pmatrix} \tag{IV.44}$$

Calcul des dérivés de y

Nous pouvons aussi obtenir les dérivées (IV.26) à (IV.28), qui se reécrivent comme

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\ \sin(\omega\tau) \\ -\mathrm{i}\ \sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \end{pmatrix} + \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} \tag{IV.45}$$

$$
\begin{aligned}
\begin{pmatrix} y_1'' \\ y_2'' \end{pmatrix} + \mathbb{A}\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} &= -\begin{pmatrix} 0 & \mathrm{i}\,\omega \\ \mathrm{i}\,\omega & 0 \end{pmatrix}\begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} + \begin{pmatrix} P_1' \\ P_2' \end{pmatrix} \\
&= -\begin{pmatrix} \omega\sin(\omega\tau) & \mathrm{i}\,\omega\cos(\omega\tau) \\ \mathrm{i}\,\omega\cos(\omega\tau) & \omega\sin(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} + \begin{pmatrix} P_1' \\ P_2' \end{pmatrix}
\end{aligned} \tag{IV.46}
$$

$$
\begin{aligned}
\begin{pmatrix} y_1''' \\ y_2''' \end{pmatrix} + \mathbb{A}\begin{pmatrix} y_1'' \\ y_2'' \end{pmatrix} &= \begin{pmatrix} -\omega^2 & 0 \\ 0 & -\omega^2 \end{pmatrix}\begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad -2\begin{pmatrix} 0 & \mathrm{i}\,\omega \\ \mathrm{i}\,\omega & 0 \end{pmatrix}\begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1''' \\ z_2''' \end{pmatrix} + \begin{pmatrix} P_1'' \\ P_2'' \end{pmatrix} \\
&= \begin{pmatrix} -\omega^2\cos(\omega\tau) & \mathrm{i}\,\omega^2\sin(\omega\tau) \\ \mathrm{i}\,\omega^2\sin(\omega\tau) & -\omega^2\cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad - \begin{pmatrix} 2\,\omega\sin(\omega\tau) & 2\,\mathrm{i}\,\omega\cos(\omega\tau) \\ 2\,\mathrm{i}\,\omega\cos(\omega\tau) & 2\,\omega\sin(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & -\mathrm{i}\,\sin(\omega\tau) \\ -\mathrm{i}\,\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix}\begin{pmatrix} z_1''' \\ z_2''' \end{pmatrix} + \begin{pmatrix} P_1'' \\ P_2'' \end{pmatrix}
\end{aligned} \tag{IV.47}
$$

Ce sont ces expressions que nous implémentons dans notre résolution numérique afin de pouvoir calculer les coefficients de notre polynôme P.

2.3.c *Equations du modèle High-Order Spectral*

En utilisant la relation de dispersion linéaire $\omega^2 = gk\,\tanh(kh)$, nous pouvons reécrire les équations vues en II.1.4.e sous la forme matricielle

$$
\frac{\partial}{\partial t}\begin{pmatrix} \hat{\eta} \\ \dfrac{\omega}{g}\hat{\phi}^s \end{pmatrix} + \mathbb{A}\cdot\begin{pmatrix} \hat{\eta} \\ \dfrac{\omega}{g}\hat{\phi}^s \end{pmatrix} = \begin{pmatrix} \hat{B}_1 \\ \dfrac{\omega}{g}\hat{B}_2 \end{pmatrix} \tag{IV.48}
$$

avec la matrice des parties linéaires $\mathbb{A} = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}$, ses puissances

$$
\mathbb{A}^2 = \begin{pmatrix} -\omega^2 & 0 \\ 0 & -\omega^2 \end{pmatrix},\ \mathbb{A}^{-1} = \begin{pmatrix} 0 & \dfrac{1}{\omega} \\ -\dfrac{1}{\omega} & 0 \end{pmatrix},\ \mathbb{A}^{-3} = \begin{pmatrix} 0 & -\dfrac{1}{\omega^3} \\ \dfrac{1}{\omega^3} & 0 \end{pmatrix} \tag{IV.49}
$$

et les exponentielles

$$
\mathrm{e}^{\mathbb{A}t} = \begin{pmatrix} \cos(\omega\ t) & -\sin(\omega\ t) \\ \sin(\omega\ t) & \cos(\omega\ t) \end{pmatrix},\qquad \mathrm{e}^{-\mathbb{A}t} = \begin{pmatrix} \cos(\omega\ t) & \sin(\omega\ t) \\ -\sin(\omega\ t) & \cos(\omega\ t) \end{pmatrix} \tag{IV.50}
$$

Relation entre les variable y et z

Comme pour les équations de *Serre*, nous obtenons la relation entre y et z

$$\begin{aligned}
\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\
&\quad - \begin{pmatrix} \cos(\omega\tau)-1 & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau)-1 \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{\omega} \\ -\frac{1}{\omega} & 0 \end{pmatrix} \begin{pmatrix} p_{0a} \\ p_{0b} \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau)-1 & \sin(\omega\tau)-\omega\tau \\ -\sin(\omega\tau)+\omega\tau & \cos(\omega\tau)-1 \end{pmatrix} \begin{pmatrix} -\frac{1}{\omega^2} & 0 \\ 0 & -\frac{1}{\omega^2} \end{pmatrix} \begin{pmatrix} p_{1a} \\ p_{1b} \end{pmatrix} \\
&\quad - \begin{pmatrix} \cos(\omega\tau)-1+\frac{1}{2}\omega^2\tau^2 & \sin(\omega\tau)-\omega\tau \\ -\sin(\omega\tau)+\omega\tau & \cos(\omega\tau)-1+\frac{1}{2}\omega^2\tau^2 \end{pmatrix} \begin{pmatrix} 0 & -\frac{1}{\omega^3} \\ \frac{1}{\omega^3} & 0 \end{pmatrix} \begin{pmatrix} p_{2a} \\ p_{2b} \end{pmatrix} \\
&= \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\
&\quad -\frac{1}{\omega} \begin{pmatrix} -\sin(\omega\tau) & \cos(\omega\tau)-1 \\ -\cos(\omega\tau)+1 & -\sin(\omega\tau) \end{pmatrix} \begin{pmatrix} p_{0a} \\ p_{0b} \end{pmatrix} \\
&\quad +\frac{1}{\omega^2} \begin{pmatrix} -\cos(\omega\tau)+1 & -\sin(\omega\tau)+\omega\tau \\ \sin(\omega\tau)-\omega\tau & -\cos(\omega\tau)+1 \end{pmatrix} \begin{pmatrix} p_{1a} \\ p_{1b} \end{pmatrix} \\
&\quad -\frac{1}{\omega^3} \begin{pmatrix} \sin(\omega\tau)-\omega\tau & -\cos(\omega\tau)+1-\frac{1}{2}\omega^2\tau^2 \\ \cos(\omega\tau)-1+\frac{1}{2}\omega^2\tau^2 & \sin(\omega\tau)-\omega\tau \end{pmatrix} \begin{pmatrix} p_{2a} \\ p_{2b} \end{pmatrix}
\end{aligned} \tag{IV.51}$$

et pour le cas où des composantes de $\mathbb{A}$ seraient nulles, nous avons la relation

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \begin{pmatrix} -\tau & 0 \\ 0 & -\tau \end{pmatrix} \begin{pmatrix} p_{0a} \\ p_{0b} \end{pmatrix} + \begin{pmatrix} \frac{\tau^2}{2} & 0 \\ 0 & \frac{\tau^2}{2} \end{pmatrix} \begin{pmatrix} p_{1a} \\ p_{1b} \end{pmatrix} - \begin{pmatrix} -\frac{\tau^3}{6} & 0 \\ 0 & -\frac{\tau^3}{6} \end{pmatrix} \begin{pmatrix} p_{2a} \\ p_{2b} \end{pmatrix} \tag{IV.52}$$

Calcul des dérivés de y

Nous pouvons aussi obtenir les dérivées (IV.26) - (IV.28), qui se reécrivent comme

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \end{pmatrix} + \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} \tag{IV.53}$$

$$
\begin{aligned}
\begin{pmatrix} y_1'' \\ y_2'' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} &= - \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} + \begin{pmatrix} P_1' \\ P_2' \end{pmatrix} \\
&= - \begin{pmatrix} \omega\sin(\omega\tau) & -\omega\cos(\omega\tau) \\ \omega\cos(\omega\tau) & \omega\sin(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} + \begin{pmatrix} P_1' \\ P_2' \end{pmatrix} \qquad \text{(IV.54)} \\
\begin{pmatrix} y_1''' \\ y_2''' \end{pmatrix} + \mathbb{A} \begin{pmatrix} y_1'' \\ y_2'' \end{pmatrix} &= \begin{pmatrix} -\omega^2 & 0 \\ 0 & -\omega^2 \end{pmatrix} \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad -2 \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1''' \\ z_2''' \end{pmatrix} + \begin{pmatrix} P_1'' \\ P_2'' \end{pmatrix} \\
&= \begin{pmatrix} -\omega^2\cos(\omega\tau) & -\omega^2\sin(\omega\tau) \\ \omega^2\sin(\omega\tau) & -\omega^2\cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1' \\ z_2' \end{pmatrix} \\
&\quad - \begin{pmatrix} 2\,\omega\sin(\omega\tau) & -2\,\omega\cos(\omega\tau) \\ 2\,\omega\cos(\omega\tau) & 2\,\omega\sin(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1'' \\ z_2'' \end{pmatrix} \\
&\quad + \begin{pmatrix} \cos(\omega\tau) & \sin(\omega\tau) \\ -\sin(\omega\tau) & \cos(\omega\tau) \end{pmatrix} \begin{pmatrix} z_1''' \\ z_2''' \end{pmatrix} + \begin{pmatrix} P_1'' \\ P_2'' \end{pmatrix} \qquad \text{(IV.55)}
\end{aligned}
$$

3 Comparaisons de simulations numériques

Pour la réalisation de nos simulations numériques développées avec le logiciel *Matlab* [2], l'ordinateur utilisé était le serveur de calcul partagé *math11* du laboratoire, possédant 12 coeurs à 2,93 GHz et 48 GBytes de RAM.

3.1 Présentation des comparaisons entre deux méthodes

Notre objectif est de comparer la taille et/ou le nombre total de pas de temps de calcul effectués pour une même simulation mais avec différentes méthodes numériques. Pour cela, il est habituel de s'intéresser au temps de calcul *CPU*. Or, pour cette thèse j'ai développé notre méthode sur un serveur partagé avec tout le laboratoire, ce qui fait que mesurer le temps *CPU* uniquement utilisé pour ma simulation n'était pas possible.
Nous allons donc voir quelle autre quantité nous pouvons étudier pour définir le gain obtenu avec notre méthode. Par contre, j'ai pu vérifier un point important. Ce qui coûte le plus cher en terme de temps *CPU* avec les équations des vagues est le calcul du terme non linéaire. Or, notre méthode a un atout majeur sur d'autres, nous ne calculons rien que nous n'ayons déjà sous une autre forme pour les termes que nous ajoutons (réutilisation des termes du Runge-Kutta pour le *Dense Output*). Pour des simulations *simples* (par exemple avec l'équation de *BBM*), le gain en terme de *CPU* est de quelques points inférieur au gain sur le nombre total de pas de temps, ce qui est normal. En effet, dès que le système simulé fait appel à une complexité de calcul plus

importante (par exemple avec les équations de *Serre* ou de *HOS*), ces deux types de gains sont équivalents. Comme je n'ai pas cherché à optimiser au mieux le temps passé dans les parties du code liées aux ajouts de notre méthode, cela confirme bien qu'avec nos modifications nous passons un temps de calcul supplémentaire négligeable dans notre méthode au regard du temps de calcul du terme non linéaire. Ainsi, plus le pas de temps sera grand, moins il en faudra pour réaliser la simulation et cela se fera sans surcoût numérique.

Exemple de simulations

Sur les figures du type IV.1 nous présentons la superposition des différents pas de calcul Δt (qui permettent d'avancer la simulation de boucles de calcul en boucles de calcul) en fonction du temps de la simulation normalisé par la période. Au début de la simulation, nous voyons que le pas de temps augmente jusqu'à atteindre un plateau. Cela s'explique par le fait que lorsque nous propageons une onde permanente solution de l'équation d'évolution, sous certaines conditions (*KdV*, *Serre*, *HOS* avec petites cambrures ...), il n'y a aucune modification du profil de l'onde. De ce fait, une fois que le mécanisme du pas de temps adaptatif se stabilise à une valeur optimale du pas de temps Δt, cette valeur restera la même pour toute la simulation, comme nous le voyons sur la figure IV.1.

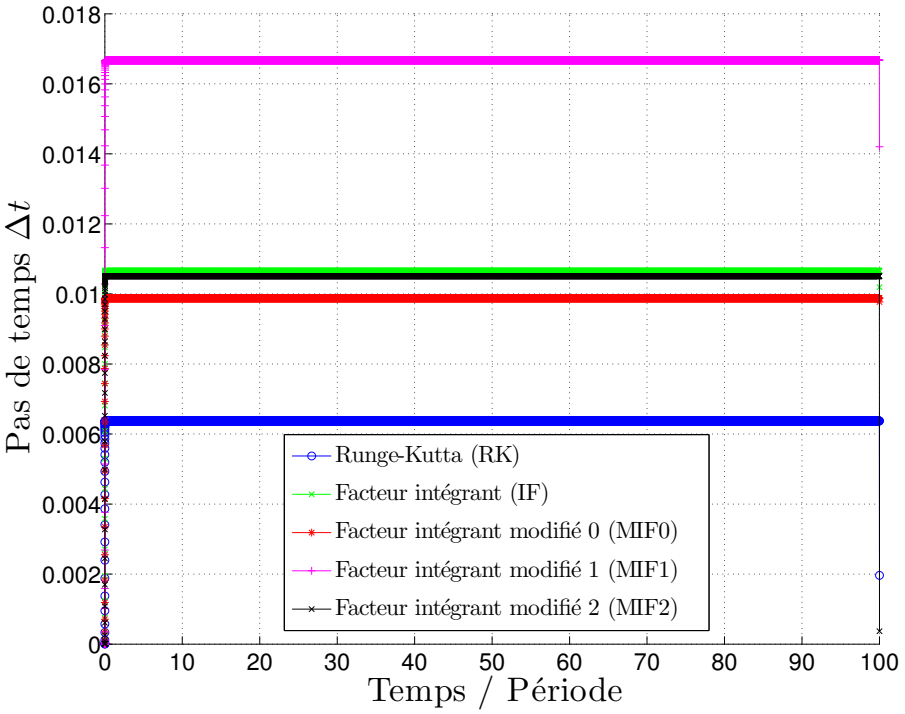

FIGURE IV.1 – Exemple d'évolution du pas de temps Δt en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge, *MIF1* en violet et *MIF2* en noir).

Cette valeur optimale ne changera que si, à certains moments, l'erreur locale se met à croître à cause d'interactions, de perturbations ou d'erreurs numériques.
Nous remarquons qu'à la fin de la simulation, à droite de la figure IV.1, il y a pour chaque méthode un point représentant la dernière valeur du pas de temps qui est plus bas que les précédents. Cela est normal puisque nous utilisons un pas de temps adaptatif, c'est-à-dire qu'en

ne fixant pas à l'avance la valeur du pas de temps, le dernier pas de temps doit avoir une taille inférieure aux autres afin de terminer la simulation au temps final prédéfini. Nous donnons les notations des différentes méthodes et leur couleurs représentatives dans les graphiques :

- Runge-Kutta classique seule, noté *RK* et en bleu.
- Runge-Kutta avec facteur intégrant classique, noté *IF* et en vert.
- Runge-Kutta avec facteur intégrant modifié à l'ordre 0, noté *MIF0* et en rouge.
- Runge-Kutta avec facteur intégrant modifié à l'ordre 1, noté *MIF1* et en violet.
- Runge-Kutta avec facteur intégrant modifié à l'ordre 2, noté *MIF2* et en noir.

Lorsque nous donnerons un gain pour la méthode avec le facteur intégrant classique *IF*, il sera calculé par rapport à la méthode du Runge-Kutta classique *RK*, tandis qu'un gain pour une méthode avec le facteur intégrant modifié *MIF* sera calculé par rapport à la méthode avec le facteur intégrant classique *IF*, cela afin de voir uniquement le gain supplémentaire apporté par notre travail par rapport à la méthode existante la plus efficace, celle du *IF*.

Les comparaisons entre ces différentes méthodes : avec ou sans facteur intégrant, seront *toujours* réalisées avec la même approche temporelle de Runge-Kutta (à savoir, soit celle de Dormand et Prince, soit celle de Bogacki et Shampine, soit celle de Verner). Par exemple, nous ne mélangerons pas des comparaisons entre une méthode de Runge-Kutta de Verner seule avec une méthode de Runge-Kutta de Bogacki et Shampine avec facteur intégrant.

Intérêt de l'étude du pas de temps

Lorsque nous travaillons avec une avance temporelle à pas de temps constant, l'outil qui permet de dire si une simulation est *meilleure* qu'une autre est souvent l'évolution de l'erreur locale en fonction du temps, du pas de temps ou d'une tolérance. Or, comme nous venons de l'expliquer, ici nous travaillons avec un pas de temps adaptatif. Cela implique que la valeur du pas de temps est directement liée à la valeur de l'erreur locale, comme nous le voyons dans les relations (II.54) et (II.56). Donc, si l'erreur locale augmente, la taille du pas de temps diminue et si l'erreur locale diminue, la taille du pas de temps augmente. De ce fait, avec un pas de temps adaptatif, le critère intéressant à regarder au cours du temps n'est plus l'erreur locale directement, mais la taille du pas de temps Δt. C'est pour cela que nous ne présentons que des graphiques du type IV.1 lors des comparaisons entre méthodes, puisque, pour ce type d'étude, toute l'information nécessaire y est directement visible.

3.2 Pourcentage de réduction et gain entre deux méthodes

Pour mesurer l'efficacité entre deux méthodes, nous pouvons regarder soit le nombre total de pas de calcul effectués, soit la valeur (constante ou moyenne selon les cas) du pas de temps Δt sur la simulation. Il est à noter que ces deux quantités sont reliées entre elles et représentent deux façons de quantifier un même effet. En effet, pour une simulation de temps total T, un pas de temps optimal constant (ou moyenné) $\Delta t_{a,b}$ pour deux méthodes a et b comparées et un nombre total $N_{a,b}$ de pas de temps, nous avons

$$\begin{aligned} \Delta t_a N_a &= T \\ \Delta t_b N_b &= T \end{aligned} \tag{IV.56}$$

c'est-à-dire

$$\frac{\Delta t_b}{\Delta t_a} = \frac{N_a}{N_b} \tag{IV.57}$$

Nous définissons deux types de mesure de la performance entre deux méthodes :
• Le *pourcentage de réduction* du nombre total de pas de calcul effectués, c'est-à-dire un taux de variation qui est

$$\text{Réduction} = 100\left(\frac{N_a - N_b}{N_a}\right) = 100\left(1 - \frac{N_b}{N_a}\right) \tag{IV.58}$$

Du point de vue notation, nous partons du fait que le pas de temps de notre nouvelle méthode, noté ici Δt_b, sera plus grand que celui de l'ancienne méthode noté Δt_a, puisque nous pensons que notre approche va augmenter la taille du pas du temps afin d'en réduire le nombre total. Donc N_b sera plus petit que N_a. Si ce n'est pas le cas, aucun problème, le pourcentage de réduction sera simplement négatif pour signifier que notre méthodologie est moins bonne que l'ancienne. Il faut noter que la limite maximale de cette indicateur est 100 %, ce qui correspond à l'obtention de la solution finale dès le premier calcul.
• Le *facteur de gain*, en terme de coefficient multiplicateur entre méthodes sur la valeur (constante ou moyenne) du pas Δt ou sur le nombre total de pas de temps, est défini par

$$\text{Gain} = \frac{N_a}{N_b} = \frac{\Delta t_b}{\Delta t_a} \tag{IV.59}$$

et permet de dire de combien le pas de temps d'une méthode est plus grand par rapport à l'autre, ou de combien une méthode se fait plus rapidement que l'autre en terme de nombres de boucles de calculs. Nous avons donc la relation suivante entre les deux types de mesures proposés ici,

$$\text{Réduction} = 100\left(1 - \frac{1}{\text{Gain}}\right) \tag{IV.60}$$

Choix de l'indicateur de performance

Pour comprendre pourquoi il est utile de faire la différence entre ces deux indicateurs, nous allons définir deux catégories de résultats. La première est celle comprenant des pourcentages de réductions compris entre 0 % et 50 %. Utiliser ces pourcentages est pertinent à l'inverse du gain qui ne serait que de l'ordre 1 à 2. La seconde catégorie est celle comprenant les poucentages de réductions beaucoup plus importants. Prenons le cas des valeurs 90 %, 95 %, 97 % ou 99 % de réduction. Ces données ne permettent pas d'apprécier les différences entre ces pourcentages, alors qu'ils représentent des gains de facteur 5, 1 000, 10 000 ou un million ... Pour cela, dans ce manuscrit nous donnerons toujours les comparaisons entre deux méthodes en terme de pourcentage de réduction, sauf pour les équations de *Serre* et *HOS*, couplées à une avance temporelle Runge-Kutta de Bogacki-Shampine, où les comparaisons seront données en terme de gain afin de mettre en lumière des écarts beaucoup plus importants entre les méthodes.

Cas d'un pas de temps oscillant

Lorsque, par exemple, nous augmentons l'amplitude ou perturbons la condition initiale, le pas de temps ne pourra pas toujours avoir la même valeur optimale et va osciller autour d'une valeur moyenne. Dans ce cas, le pourcentage de réduction sur le nombre total de boucles de calcul effectuées est toujours une mesure *cohérente*, tout comme pour le facteur de gain, pour lequel il faut prendre en compte, non plus la valeur constante du pas de de temps Δt mais sa valeur moyenne (ou alors le nombre total de pas de temps). Pour le vérifier, prenons l'exemple de la simulation *HOS* à la section VII.2.2 dont l'évolution des pas de temps se trouve sur la

figure VII.4. Il s'avère que le gain basé sur le nombre total de pas de temps est bien identique à celui basé sur la valeur moyenne de Δt, comme en témoigne le tableau comparatif IV.1.

TABLE IV.1 – Comparaison des facteurs gains entre deux méthodes, sur le nombre total de pas de temps et sur la valeur moyenne du pas de temps.

Avec le nombre total de pas de temps	**Avec la valeur moyenne de Δt**
4.24	4.24
26.26	26.26

4 DISCUSSION

Nous avons vu, d'une part comment nous pouvons modifier le facteur intégrant classique en ajoutant à une équation d'évolution un certain polynôme P, qui peut être développé à différents ordres, et d'autre part son application à certaines équations des vagues. Nous avons aussi appris à utiliser différentes avances temporelles de type Runge-Kutta. En liant ces deux aspects d'intégration temporelle, nous pensons pouvoir économiser un temps de calcul non négligeable lors des simulations numériquement difficiles en augmentant la taille des pas de temps et donc en diminuant le nombre total de boucles de calculs nécessaires à une simulation. Nous allons maintenant regarder les résultats numériques obtenus avec notre méthode pour des simulations de vagues extrêmes. En premier lieu, nous testons nos schémas numériques avec des équations simples à simuler afin de vérifier leur bon fonctionnement.

Chapitre V

Applications aux équations de *KdV*, *BBM* et *NLS*

Nous allons nous pencher sur les premiers types d'équations de vagues qui nous intéressent. Tout d'abord celle de Korteweg et de Vries (voir I.2.1), notée *KdV*, puis celle de Benjamin, Bona et Mahony (voir I.2.2), notée *BBM* et enfin celle de Schrödinger Non Linéaire, notée *NLS* (voir I.2.3).

Pour nous, ces équations ne sont que des *équations d'essais* nous permettant de tester l'implémentation des différentes méthodes. En effet, comme expliqué en II.3, notre facteur intégrant modifié fait appel à des assemblages de données supplémentaires par rapport au facteur intégrant classique. Cela nécessite un coût en temps de calcul non négligeable pour ces équations, mais négligeable pour des équations possédant des termes non linéaires plus difficiles à manipuler, comme pour les modèles de *Serre* (chapitre VI) et de *HOS* (chapitre VII). Aussi, dans ce chapitre, nous n'étudions pas les trois premières équations *sous toutes leurs coutures*, mais ne donnons que l'essentiel pour nous, à savoir, est-ce que le facteur intégrant modifié fonctionne et dans quelles conditions ? Des études plus poussées seront donc réalisées pour les équations des chapitres suivants.

1 Equation de Korteweg et de Vries

Cette section contient les résultats des simulations pour la méthode temporelle de Dormand et Prince.

1.1 Méthode temporelle de Dormand et Prince

Onde cnoïdale

Pour cette simulation nous partons de la solution cnoïdale vue en I.2.1 puis nous laissons évoluer le système pendant 100 périodes temporelles. Nous prenons le profil d'onde défini par les paramètres suivants : $N = 256$ le nombre de points, $m = 0.99$ le paramètre cnoïdal, $H = 0.5$ l'amplitude de l'onde et une tolérance de 10^{-12}.
Sur la figure V.1 représentant l'évolution du pas de temps de calcul Δt en fonction du temps normalisé par la période, nous voyons qu'à l'aide de la méthode *MIF0* nous obtenons une réduction du nombre total de pas de temps d'environ 51% par rapport à la méthode *IF*. Cela veut donc dire qu'il faut 2 fois moins de temps de calcul pour réaliser une même simulation.

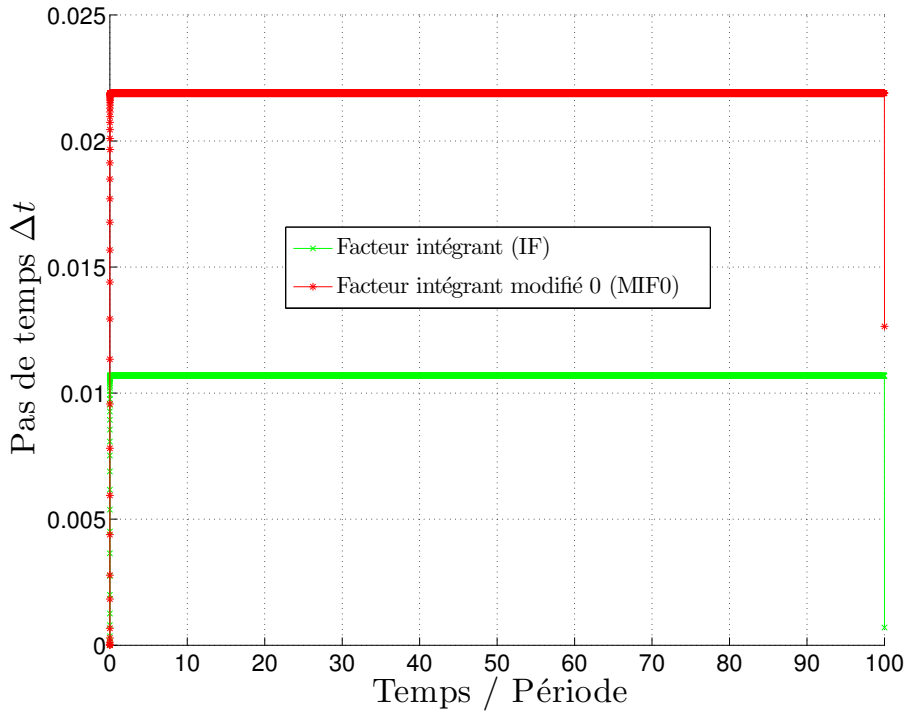

FIGURE V.1 – Evolution du pas de temps $\Delta\ t$ en fonction du temps de simulation normalisé par la période pour deux schémas numériques (*IF* en vert et *MIF0* en rouge). Les paramètres sont : tolérance à 10^{-12}, $H = 0.5$ et $m = 0.99$.

En faisant de même avec un soliton comme profil initial pour une même amplitude, nous obtenons une réduction de l'ordre de 25 % en nombre total de pas de temps de calcul pour notre méthode *MIF0* par rapport à la méthode *IF*.

Remarques

Seules les méthodes *IF* et *MIF0* nous permettent d'obtenir un résultat. Celles d'ordre supérieur *MIF1* et *MIF2* accumulent trop d'erreurs numériques dès le départ et donc le pas de temps *s'écroule*, rendant les simulations inefficaces, comme l'a aussi remarqué Krogstad ([50] page 86) pour son facteur intégrant généralisé aux ordres 1 et 2.

Nous obtenons des résultats similaires avec le facteur intégrant généralisé, puisqu'à l'ordre 0 nos deux facteurs intégrants modifié et généralisé sont équivalents et aux ordres supérieurs les erreurs numériques nous empêchent de terminer les simulations.

1.2 Discussion

Sur ces premiers cas de simulations de vagues, nous avons pu observer que le facteur intégrant modifié semble être plus avantageux que le facteur intégrant classique. Maintenant nous allons pouvoir étudier plus précisément le comportement de nos différents intégrateurs exponentiels avec l'équation suivante. En effet, avec l'équation de *KdV*, ni la méthode de Runge-Kutta classique ni les méthodes *MIF1* et *MIF2* ne permettent de propager une onde sans erreurs numériques. Afin de pouvoir réaliser une étude plus complète il est donc préférable d'utiliser une équation proche de celle de *KdV*, l'équation de *BBM*, qui, à l'aide d'une légère modification, permet de réaliser ces simulations, comme expliqué en I.2.2.

2 Equation de Benjamin Bona et Mahony

Cette section contient les résultats des simulations pour les méthodes temporelles de Dormand et Prince en V.2.1, de Bogacki et Shampine ainsi que de Verner en V.2.2.

Le facteur intégrant modifié et le facteur intégrant généralisé à l'ordre 2 ne peuvent nous donner des résultats concluants pour cause de bruit numérique trop important.

2.1 Méthode temporelle de Dormand et Prince

Pas de temps Δt

Avant de faire un tour d'horizon des différents jeux de paramètres possibles, nous allons regarder en détail un cas précis pour l'avance temporelle de Dormand et Prince. Nous partons de la solution cnoïdale, puis nous laissons évoluer le système pendant 100 périodes de notre boîte de calcul, à l'aide des différents codes *RK*, *IF*, *MIF0*, *MIF1* et *MIF2*. Une fois la simulation terminée, nous renversons le temps afin de revenir à l'état initial. Nous fixons les paramètres, à savoir : l'amplitude $H = 1.5$, le nombre de points $N = 256$, l'onde cnoïdale définie par $m = 0.99$ et la tolérance à 10^{-12}. Sur la figure V.2 nous avons l'évolution du pas de temps en fonction du nombre de périodes temporelles effectuées. Les courbes du haut représentent l'évolution pour un temps positif, c'est-à-dire lorsque nous partons de la condition initiale et laissons évoluer le système jusqu'à la centième période. Une fois cette première simulation terminée, nous renversons le temps pour retourner de la centième période à la condition initiale. Dans ce cas, le pas de temps sera négatif et nous obtenons les courbes inférieures. Nous voyons qu'une

fois que le pas de temps optimal est trouvé au début de la simulation, celui-ci reste constant du fait que nous faisons se propager une onde progressive sans perturbations. Lorsque nous renversons le temps, nous retrouvons la même valeur du pas de temps au signe près, ce qui nous permet de dire que les schémas numériques sont réversibles. Le fait que le pas de temps adaptatif ne varie pas est déjà une première indication que dans la simulation il n'y a pas de problèmes d'erreurs numériques ou autres, sinon le pas de temps décroîtrait puisqu'il est directement lié à l'erreur locale de la simulation.

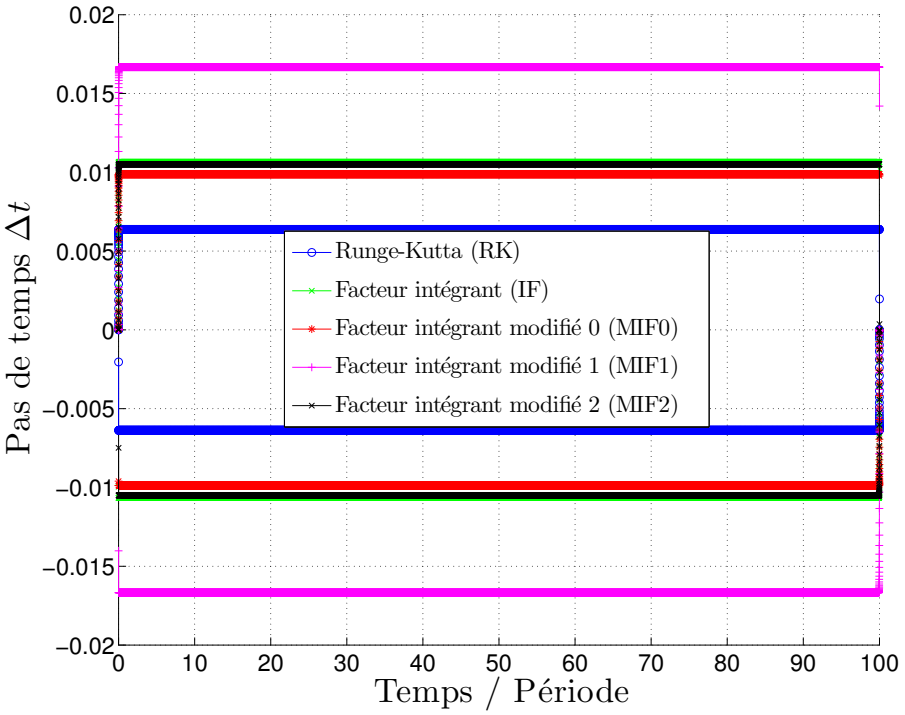

FIGURE V.2 – Evolution du pas de temps Δt en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge, *MIF1* en violet et *MIF2* en noir). Les paramètres sont : tolérance à 10^{-12}, $H = 1.5$ et $m = 0.99$.

Dans le cas de ce jeu de paramètre, il apparaît que le facteur intégrant classique permet une réduction de 40 % sur le nombre total de pas de temps par rapport à la méthode de Runge-Kutta classique. Nous voyons aussi que notre méthode de facteur intégrant modifié à l'ordre 0 fait moins bien, environ 8%, que le facteur intégrant classique. En revanche, à l'ordre 1 notre méthode de facteur intégrant modifié est meilleure que le facteur intégrant classique puisque nous pouvons réduire le nombre de calculs de plus de 36 %. Enfin, à l'ordre 2 nous reperdons l'avantage sur le facteur intégrant classique, à cause de problèmes d'erreurs numériques comme dit précédemment.

Tableau de pourcentages de réduction entre les méthodes

A l'aide de l'avancement temporel de type Runge-Kutta de Dormand et Prince, nous faisons varier le profil de l'onde en modifiant à la fois le paramètre m de l'onde cnoïdale et l'amplitude H. Nous fixons le nombre de points $N = 256$ et la tolérance à 10^{-12}. Le tout est fait pour une avance temporelle de 100 périodes et les pourcentages de réductions du nombre de pas de temps entre méthodes sont regroupés dans le tableau V.1. Le symbole « x » signifie que toutes les méthodes sont touchées par des erreurs numériques importantes, ce qui a pour conséquence

d'imposer un pas de temps de plus en plus petit, et donc un temps de simulation qui ne cesse d'augmenter à chaque calcul. De ce fait, ces simulations ne sont pas réalisables.
Nous voyons dans ce tableau que si le facteur intégrant modifié à l'ordre 0 ne fait pas mieux que le facteur intégrant classique, tout comme pour celui à l'ordre 2, en revanche le facteur intégrant modifié à l'ordre 1 nous permet, selon les paramètres, d'obtenir au maximum des réductions du nombre total de pas de temps de l'ordre de 35 % sur le facteur intégrant classique.

TABLE V.1 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0*, *MIF1* et *MIF2* sont définies par rapport à la méthode *IF* classique.

m		**0.9999**	**0.99**	**0.8**	**0.6**	**0.4**	**0.2**	**0.1**
H=0.1	**IF**	85.55	86.44	86.71	85.78	84.07	82.53	78.82
	MIF0	- 114.81	- 124.07	- 121.57	- 104.92	- 78.12	- 31.45	3.24
	MIF1	- 172.17	- 185.09	- 184.27	- 164.64	- 134.13	- 88.00	- 25.17
	MIF2	- 177.60	- 191.58	- 193.63	- 176.11	- 149.23	- 111.18	- 41.16
H=0.2	**IF**	76.64	78.14	79.32	78.60	77.75	75.61	72.67
	MIF0	- 51.61	- 57.13	- 58.75	- 49.67	- 32.33	- 1.12	- 0.35
	MIF1	- 85.80	- 95.11	- 100.08	- 90.18	- 73.35	- 27.58	- 1.19
	MIF2	- 95.08	- 104.95	- 112.09	- 104.50	- 92.51	- 47.48	- 2.65
H=0.4	**IF**	64.66	66.86	69.22	69.50	70.57	66.97	x
	MIF0	- 22.12	- 24.53	- 25.73	- 20.21	- 10.91	- 1.60	
	MIF1	- 30.29	- 37.30	- 43.05	- 38.84	- 27.78	4.04	
	MIF2	- 45.54	- 52.34	- 59.17	- 57.45	- 51.25	- 11.70	
H=0.6	**IF**	56.46	59.06	62.29	63.97	64.00	x	x
	MIF0	- 14.53	- 15.91	- 16.19	- 12.62	- 6.07		
	MIF1	- 5.41	- 11.73	- 17.76	- 16.40	1.94		
	MIF2	- 26.09	- 31.61	- 37.97	- 39.25	- 27.28		
H=0.8	**IF**	50.34	53.19	57.23	59.75	58.83	x	x
	MIF0	- 11.35	- 12.32	- 12.13	- 9.68	- 4.60		
	MIF1	10.75	4.62	- 2.25	- 0.48	16.26		
	MIF2	- 15.25	- 20.06	- 26.41	- 28.16	- 15.41		
H=1.0	**IF**	45.53	48.54	53.37	56.04	54.85	x	x
	MIF0	- 9.58	- 10.38	- 10.06	- 8.21	- 3.93		
	MIF1	23.09	17.09	8.82	13.24	17.06		
	MIF2	- 8.21	- 12.53	- 19.05	- 19.91	- 8.96		

H=1.2	**IF**	41.64	44.75	50.31	52.72	x	x	x
	MIF0	- 8.43	- 9.14	- 8.90	- 7.30			
	MIF1	32.13	27.38	17.28	24.31			
	MIF2	- 3.20	- 7.17	- 13.83	- 13.53			
H=1.4	**IF**	38.40	41.59	47.76	49.78	x	x	x
	MIF0	- 7.59	- 8.26	- 8.19	- 6.67			
	MIF1	35.24	34.62	23.53	28.41			
	MIF2	0.57	- 3.12	- 9.84	- 8.59			
H=1.6	**IF**	35.66	38.90	45.59	47.20	x	x	x
	MIF0	- 6.95	- 7.60	- 7.74	- 6.19			
	MIF1	33.99	36.05	27.26	26.97			
	MIF2	3.53	0.06	- 6.61	- 4.71			

L'autre enseignement que nous pouvons tirer de ces expériences numériques est que notre méthode est de plus en plus intéressante lorsque nous augmentons l'amplitude H ou diminuons le paramètre m (donc lorsque nous nous éloignons d'un signal variant peu pour tendre vers un signal sinusoïdal). Ainsi, il semblerait que plus le profil de l'onde subit de *variations* ou que plus l'amplitude de l'onde est grande (c'est-à-dire que plus il devient difficile de calculer le membre non linéaire à droite de l'équation de propagation), plus notre intégrateur exponentiel modifié est avantageux vis à vis du facteur intégrant classique.
Enfin, nous pouvons constater que notre facteur intégrant modifié devient intéressant lorsque le facteur intégrant classique perd de son efficacité, c'est-à-dire par exemple lorsqu'en augmentant l'amplitude, le facteur intégrant classique passe d'une réduction du nombre de pas de temps de 85 % à 35 % sur la méthode de Runge-Kutta classique, là, où, dans le même temps, notre facteur intégrant modifié à l'ordre 1 passe d'une réduction négative (donc un ajout de pas de calculs) de -170 % à 34 % sur le facteur intégrant classique. Cela nous laisse présager de bons résultats avec des équations encore plus fortement non linéaires.
En faisant de même pour des tolérances plus grandes ($10^{-10}, 10^{-8}$ ...), nous obtenons très peu de variations dans les données.

2.2 Résultats avec les autres méthodes pour les mêmes paramètres

Méthode de Bogacki et Shampine

Avec cette méthode d'ordre peu élevé, le facteur intégrant modifié à l'ordre 0 nous apporte une réduction maximum de 10 %, ce qui fait que le facteur intégrant classique reste suffisant. Pour rappel, les ordres supérieurs de notre méthode ne sont pas accessibles pour cette avance temporelle pour cause de limitation du Dense Output (voir IV.1.4).

Méthode de Verner

En utilisant la méthode d'avance temporelle d'ordre beaucoup plus élevé de Verner, le facteur intégrant classique s'avère être toujours plus intéressant que le facteur intégrant modifié. Sauf à moyennes et fortes amplitudes où les deux méthodes sont identiques avec le facteur intégrant modifié à l'ordre 0. Tandis qu'à l'ordre 1, le facteur intégrant modifié fait à peine mieux que le facteur intégrant classique avec 5% de réduction. Enfin, la méthode à l'ordre 2 est toujours touchée par un bruit numérique important, nous empêchant d'être plus performant que le facteur intégrant classique.

Adaptation du facteur intégrant généralisé

En adaptant le facteur intégrant généralisé de Krogstad (cf. III.4.2.b), nous obtenons des résultats similaires à notre méthode de facteur intégrant modifié dans tous les cas précédents.

Dans le cas de la méthode de bas ordre de Bogacki et Shampine, contrairement au facteur intégrant modifié qui ne peut dépasser l'ordre 0 à cause du Dense Output utilisé, il est possible d'utiliser les ordres 1 et 2 du facteur intégrant généralisé. Néanmoins, l'ordre 1 ne nous fait rien gagner de plus que le facteur intégrant modifié à l'ordre 0 et l'ordre 2 est toujours trop bruité. En résumé, le facteur intégrant généralisé n'est pas à retenir. Donc, le fait que notre facteur intégrant modifié ne puisse pas atteindre ces ordres avec cette méthode de Runge-Kutta n'est pas un point négatif.

2.3 Discussion

Nous venons de voir qu'en variant à la fois la tolérance du pas de temps adaptatif et les paramètres de l'onde cnoïdale, nos intégrateurs exponentiels modifiés peuvent être plus avantageux en réduisant sensiblement (jusqu'à près de 40 %) le nombre total de pas de temps par rapport au facteur intégrant classique. Ce qui est d'autant plus intéressant, c'est qu'il semble que plus nous avons un calcul numérique difficile à réaliser, plus le facteur intégrant modifié est bénéfique, là où par contre le facteur intégrant classique montre ses limites en perdant de son intérêt. Pour continuer dans cette voie d'étude, nous allons nous intéresser à un autre modèle de vagues afin de voir ce qui se passe avec une autre équation courante, celle de *Schrödinger Non Linéaire*.

3 Equation de Schrödinger Non Linéaire

Les résultats des simulations présentés ici sont ceux obtenus par les méthodes temporelles de Dormand et Prince en V.3.1, de Bogacki et Shampine ainsi que de Verner en V.3.2.

Pour les trois méthodes temporelles de Runge-Kutta étudiées, tant pour notre facteur intégrant modifié que pour le facteur intégrant généralisé de Krogstad, le polynôme ne peut dépasser l'ordre zéro (*MIF0*) pour cause de bruit numérique trop important dès l'ordre 1.

3.1 Méthode temporelle de Dormand et Prince

Petites et moyennes amplitudes

Pour les petites et moyennes amplitudes, nous avons le récapitulatif ci-dessous (tableau V.2) des résultats pour une évolution sur 100 périodes temporelles d'une période d'onde, en fonction de l'amplitude initiale a, de l'amplitude complexe de la porteuse ψ de la surface libre η (voir I.2.3) et de la taille de la boîte de calcul L. Cette dernière est choisie de manière à ce qu'au bord du domaine considéré, le profil de l'onde tende vers le zéro machine pour éviter toute propagation d'erreurs numériques sur la condition initiale. Le tout est réalisé avec 2048 points. Nous observons que pour un repère fixe, le facteur intégrant classique est nettement supérieur à la méthode de Runge-Kutta classique (avec des pourcentages de réduction dépassant les 94 %), et que notre méthode n'a aucun intérêt et ne peut donc pas rivaliser, du moins à faible amplitude. Car si nous prêtons attention aux pourcentages de réduction, même négatifs, nous nous apercevons que plus l'amplitude augmente et avec elle la difficulté numérique de calcul, moins le facteur intégrant classique est bon et plus notre facteur intégrant modifié s'améliore.

TABLE V.2 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0* sont définies par rapport à la méthode *IF* classique.

a		**0.01**	**0.1**
L		**6000**	**600**
Repère fixe	**IF**	99.30	94.78
	MIF0	- 1733.96	- 197.16
Repère mobile	**IF**	93.96	90.86
	MIF0	30.00	30.32

En revanche, pour un repère mobile, même si pour le facteur intégrant classique nous obtenons des pourcentages de réduction conséquents (dépassant les 90 %), avec notre facteur intégrant modifié nous avons 30 % de réduction du nombre total de boucles de calculs supplémentaires. En faisant de même pour des tolérances plus grandes ($10^{-10}, 10^{-8}$ ...), nous obtenons très peu de variations dans les données.

Grandes et très grandes amplitudes

Dans le cas des grandes et très grandes amplitudes (qui physiquement ne correspondent plus à grand chose mais qui numériquement nous permettent toujours de comparer les différentes méthodes entre elles), nous faisons des simulations uniquement sur 20 périodes temporelles. En effet, en dépassant ce nombre de périodes, des erreurs numériques importantes se présentent pour toutes les méthodes.

TABLE V.3 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0* sont définies par rapport à la méthode *IF* classique.

a		**1**	**10**
L		**60**	**6**
Repère fixe	**IF**	86.62	77.51
	MIF0	11.95	30.58
Repère mobile	**IF**	85.61	77.35
	MIF0	30.39	30.94

Dans le cas du repère fixe nous voyons nettement que le facteur intégrant classique perd de l'intérêt au fur et à mesure que nous augmentons l'amplitude du profil, tandis qu'à l'inverse, notre technique de facteur intégrant modifié nous permet d'obtenir jusqu'à 30 % de réduction de pas de temps. Dans le cas du repère mobile, nous obtenons des résultats similaires aux faibles amplitudes, à savoir un pourcentage de réduction constant de l'ordre de 30 % sur le nombre total de boucles de calculs nécessaires pour une même simulation par rapport au facteur intégrant classique. En faisant de même pour des tolérances plus grandes ($10^{-10}, 10^{-8}$ ...), nous obtenons très peu de variations dans les données.

3.2 Résultats avec les autres méthodes pour les mêmes paramètres

Méthode de Bogacki et Shampine

Avec cette méthode de faible ordre, pour un repère fixe nous pouvons augmenter notre pourcentage de réduction jusqu'à 40 % à l'amplitude maximum de nos tests, contre 30 % ci-dessus. Par contre, pour un repère mobile nous obtenons une réduction de l'ordre de 70 % pour le facteur intégrant modifié à l'ordre 0 et ce avec toutes les amplitudes étudiées. Si nous obtenons un si bon résultat c'est parce que le facteur intégrant classique ne permet plus que des pourcentages de réductions de 40 % sur la méthode de Runge-Kutta classique, contre 90 % ci-dessus. Ainsi, nous voyons une fois de plus que notre méthode semble être plus avantageuse lorsque le facteur intégrant classique perd de son efficacité.

Méthode de Verner

En utilisant la méthode d'avance temporelle d'ordre beaucoup plus élevé de Verner nous n'avons aucune réduction avec nos méthodes et ce, quelque soit le jeu de paramètres employés.

Adaptation du facteur intégrant généralisé

Le facteur intégrant généralisé de Krogstad (cf. III.4.2.b) donne des résultats similaires à notre méthode de facteur intégrant modifié dans tous les cas précédents et subit les mêmes limitations sur l'ordre utilisable de la méthode (uniquement l'ordre 0).

3.3 Discussion

Nous venons de voir que nous sommes capables de faire économiser au minimum 30 % de boucles de calculs lors d'une simulation à l'aide d'un schéma d'ordre moyen (5 pour Dormand et Prince) et plus de 70 % avec une avance temporelle de faible ordre (3 pour celui de Bogacki et Shampine). Ces premiers résultats *tests* sont encourageants pour la suite, c'est-à-dire pour passer à des équations beaucoup plus difficiles à simuler en compliquant le calcul du membre non linéaire.

Avant d'attaquer le modèle totalement non linéaire, nous allons regarder le cas des équations couplées de *Serre* que nous allons étudier plus en détails.

Chapitre VI

Application aux équations de *Serre*

Après avoir étudié rapidement les cas d'une seule équation modèle dans le chapitre précédent, afin de tester les différentes approches sur des *équations tests*, nous nous intéressons ici au cas de deux équations d'évolutions couplées, à l'aide du modèle de *Serre*, vu en I.2.4. Dans ce cas d'étude, les manipulations numériques introduites par notre facteur intégrant modifié sont négligeables en temps de calcul par rapport au temps nécessaire pour calculer le terme non linéaire. De ce fait, nous étudions les résultats de ce modèle plus en détails que précédemment. Nous présentons ici les résultats des simulations pour les méthodes temporelles de Dormand et Prince en VI.1 et de Bogacki et Shampine en VI.2. En VI.3 nous faisons de même pour la méthode temporelle de Verner ainsi que pour l'emploi du facteur intégrant généralisé au lieu de notre facteur intégrant modifié. Pour finir, en VI.4 nous comparons les erreurs numériques commises entre toutes les méthodes.

1 Méthode temporelle de Dormand et Prince

Lois de conservations

Pour une simulation d'une période d'onde cnoïdale ayant pour paramètres $m = 0.99$ et $a = 0.20$, nous faisons évoluer le système sur 50 périodes temporelles, puis nous renversons le temps pour revenir à la condition initiale. Le tout avec une tolérance de 10^{-8} et $N = 256$ points. Sur les figures VI.1 et VI.2 nous traçons l'évolution des quantités qui devraient être conservées, à savoir la masse, la quantité de mouvement, l'énergie et la vorticité potentielle. Quelque soit la méthode utilisée (avec ou sans facteur intégrant), nous constatons que ces quantités suivent la même évolution dans les mêmes ordres de grandeurs, avant et après le retournement du temps, symbolisé par le trait noir vertical au temps intermédiaire de 50 périodes. Cela nous permet de dire que toutes ces méthodes sont valables les unes vis à vis des autres.

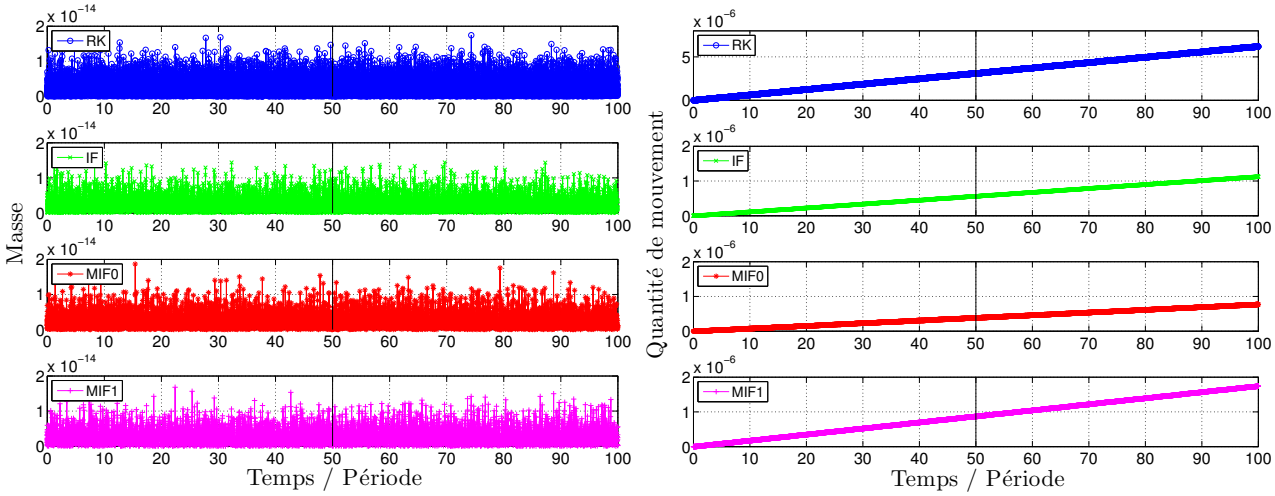

FIGURE VI.1 – Conservations de la masse (gauche) et de la quantité de mouvement (droite) en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet). Les paramètres sont : tolérance à 10^{-8}, $a = 0.2$ et $m = 0.99$.

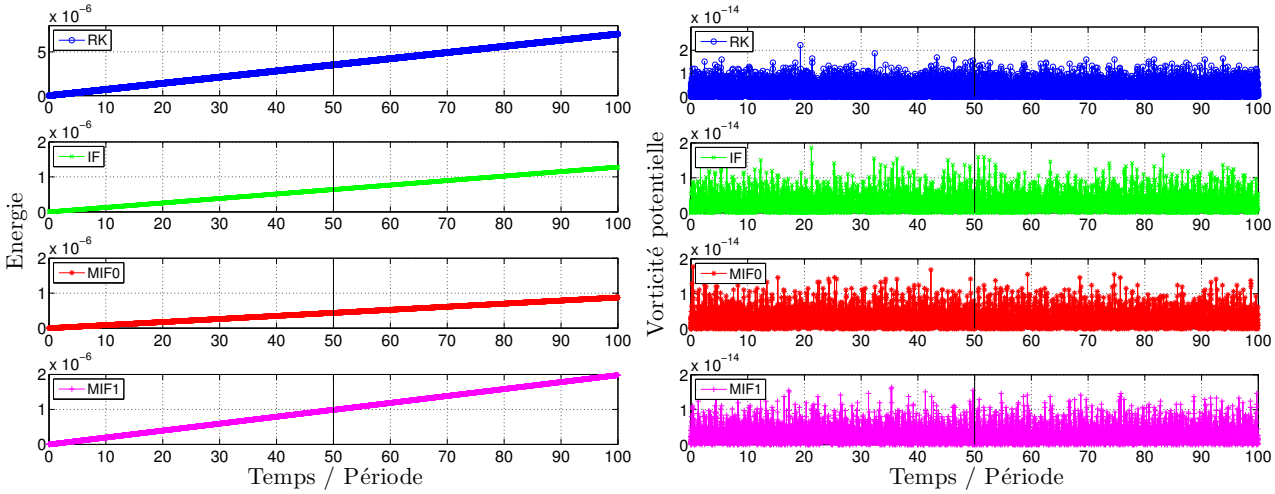

FIGURE VI.2 – Conservations de l'énergie (gauche) et de la vorticité potentielle (droite) en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet). Les paramètres sont : tolérance à 10^{-8}, $a = 0.2$ et $m = 0.99$.

Déphasage de l'onde

Comme nous connaissons la solution analytique de l'onde cnoïdale, nous pouvons en comparer les profils exact et numérique pour évaluer l'erreur commise. Pour cela, nous repérons la position du maximum de l'onde à chaque instant et regardons la différence (ou déphasage) avec la position exacte. L'exemple ci-dessous est un extrait d'une simulation (de paramètres $m = 0.99$, $a = 0.20$, de tolérance 10^{-8} et avec $N = 256$ points) pour laquelle nous avons fait un avancement temporel de 50 périodes. Quelque soit l'intervalle de temps choisi, nous obtenons toujours un graphique similaire à la figure VI.3, obtenu ainsi pour plusieurs méthodes (*RK*, *IF*, *MIF0* et *MIF1*). Pour cette simulation, le pas de discrétisation dans l'espace physique est

$\Delta x = 0.0644$. Nous constatons que l'erreur sur la position vaut au maximum $\frac{\Delta x}{2}$. Cela est cohérent avec le fait que, si notre schéma numérique est correct, nous devons faire une erreur maximum sur la phase de l'ordre du demi-pas de discrétisation.
A noter que cela est aussi valable pour les autres schémas numériques d'avance temporelle utilisés dans les autres sections de ce chapitre.

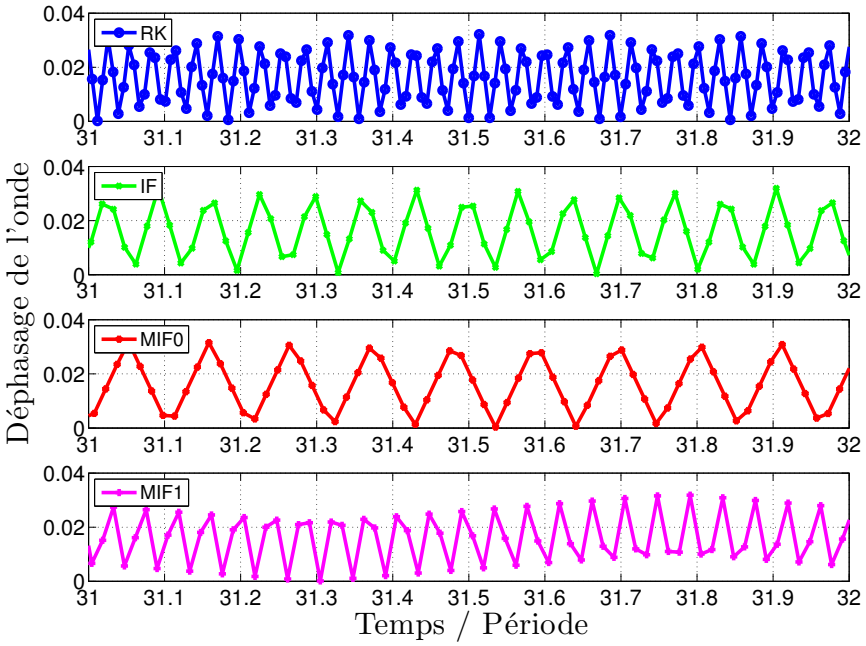

FIGURE VI.3 – Déphasage entre la solution exacte et la solution simulée en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet).

Tableau des pourcentages de réduction entre les méthodes

Ci-dessous nous faisons varier le paramètre m de l'onde cnoïdale, en fonction de différentes amplitudes, pour une tolérance fixée à 10^{-12} et pour une évolution de 10 périodes temporelles. Nous obtenons des résultats similaires pour des tolérances plus fortes.
Chaque valeur du tableau VI.1 correspond au pourcentage de réduction en terme de nombre total de pas de temps Δt. Le symbole « x » signifie que la méthode est affectée par des erreurs numériques importantes, ce qui a pour conséquence d'imposer un pas de temps de plus en plus petit et donc un temps de simulation qui ne cesse d'augmenter. Cela indépendamment du solveur temporel utilisé.
Sur le tableau suivant, nous remarquons que plus nous diminuons le paramètre m (pour faire tendre le profil de l'onde vers une sinusoïde et donc vers un maximum d'oscillations), ou que nous augmentons l'amplitude a, plus notre facteur intégrant modifié devient performant, jusqu'à atteindre une réduction de nombre de boucles de calculs de 20 % sur le facteur intégrant classique.

TABLE VI.1 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0* et *MIF1* sont définies par rapport à la méthode *IF* classique.

m		**0.9999**	**0.999**	**0.99**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**
a=0.1	**IF**	63.04	62.34	61.07	58.07	56.08	54.12	52.21	50.45	47.59	41.50	34.18
	MIF0	- 2.71	- 2.48	- 2.03	- 1.26	- 0.81	- 0.32	0.19	0.47	0.36	0.08	- 0.06
	MIF1	- 15.91	- 14.66	- 12.55	- 8.66	- 6.76	- 5.14	- 4.09	- 3.64	- 1.86	2.27	3.94
a=0.2	**IF**	56.99	55.89	53.93	49.40	46.41	43.71	41.06	36.69	29.06		
	MIF0	4.25	4.23	3.96	2.98	2.40	2.00	1.48	0.95	0.96	x	x
	MIF1	- 9.63	- 7.68	- 4.62	0.43	2.31	3.11	3.52	5.91	10.22		
a=0.3	**IF**	43.58	42.48	40.54	36.23	33.43	30.75	26.87	19.90			
	MIF0	4.10	3.88	3.50	2.57	2.05	1.69	1.47	1.79	x	x	x
	MIF1	9.09	9.72	10.56	11.15	10.84	10.56	11.98	17.31			
a=0.4	**IF**	34.12	33.09	31.23	27.15	24.41	21.19	15.27				
	MIF0	2.89	2.80	2.58	2.05	1.78	1.75	2.21	x	x	x	x
	MIF1	14.03	14.27	14.49	14.41	14.27	15.25	20.69				

2 Méthode temporelle de Bogacki et Shampine

C'est en faisant appel à une méthode d'ordre peu élevé que nous avons trouvé des résultats beaucoup plus spectaculaires, comme le laissaient supposer ceux de l'équation de *NLS*. En effet, il est possible d'obtenir des pourcentages de réductions proches des 98 % avec le facteur intégrant modifié sur le facteur intégrant classique, ce dernier ne faisant pas mieux que l'approche de Runge-Kutta seule.
Comme expliqué en IV.3.2, puisque, obtenir un pourcentage de réduction entre deux méthodes de l'ordre de 80%, 90% ou 92% semble équivalent, nous allons ici donner le facteur gain (en terme de nombre total de pas de temps) entre les méthodes pour montrer les grands écarts créés. En effet, une réduction de 80% correspond à un facteur gain 5, une réduction de 90% représente un facteur gain 10 et une réduction de 92% un facteur gain 18.

Pas de temps Δt

Pour le cas d'amplitude $a = 0.1$, de paramètre $m = 0.6$ à la tolérance 10^{-12} et pour $N = 256$, nous pouvons tracer les différents pas de temps sur la figure VI.4.

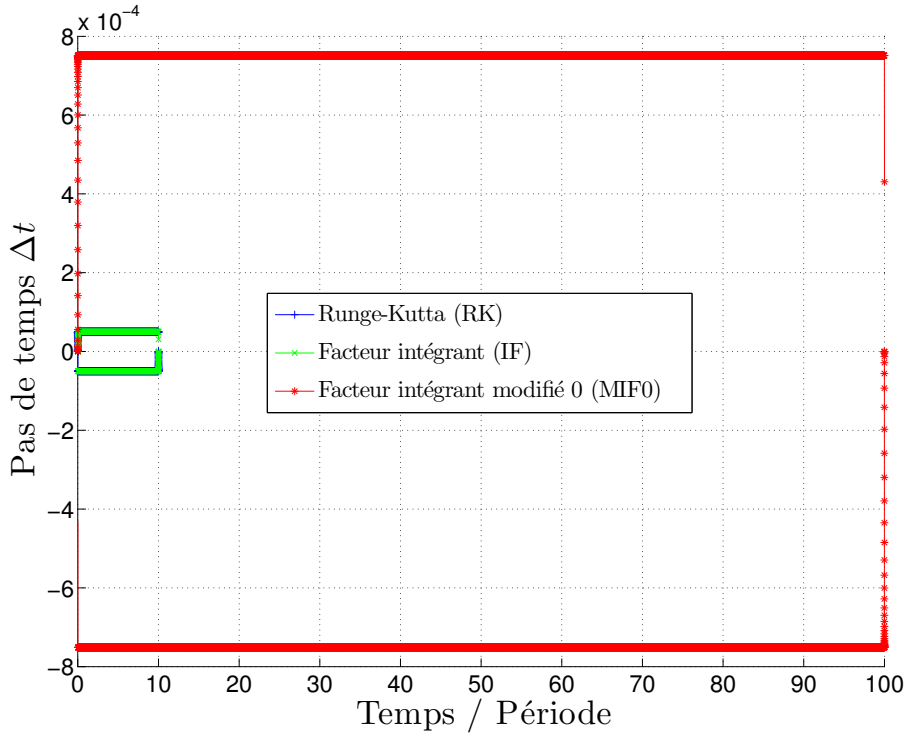

FIGURE VI.4 – Evolution du pas de temps Δt en fonction du temps de simulation normalisé par la période, pour un aller (courbes hautes) et un retour (courbes basses) de l'onde initiale sur 10 ou 100 périodes. Le tout est réalisé pour différents schémas numériques (*RK* en bleu, *IF* en vert et *MIF0* en rouge. Les paramètres sont : tolérance à 10^{-12}, $a = 0.1$ et $m = 0.6$.

Nous avons fait une simulation sur 100 périodes puis avons retourné le temps pour revenir à la condition initiale pour la méthode du facteur intégrant modifié. Par contre, pour les méthodes de Runge-Kutta classique et de facteur intégrant classique, nous n'avons une évolution temporelle

que sur 10 périodes aller et 10 périodes retour pour cause de pas de temps trop petit et donc de temps de calcul trop grand. Dans le cas de cette simulation, le facteur intégrant classique n'apporte aucun gain par rapport à la méthode de Runge-Kutta classique. Par contre, le facteur intégrant modifié à l'ordre 0 apporte un facteur gain de 15.04 sur le nombre de pas de temps par rapport au facteur intégrant classique.

Lois de conservations

Sur la figure VI.5 nous avons les évolutions au cours du temps de deux quantités qui devraient être conservées, pour des simulations aller et retour. La démarcation du renversement du temps étant le trait noir vertical au milieu du graphique. D'ailleurs, à cet instant, il est parfois possible de voir les quantités varier fortement puisque nous repartons pour une nouvelle simulation, donc avec un pas de temps très petit (et donc des quantités très précises) avant d'atteindre la valeur optimale du pas de temps.
Il apparaît que selon les quantités, les ordres de grandeurs et l'évolution de ces quantités qui doivent être conservées sont à l'avantage de la méthode du facteur intégrant modifié, puisqu'avec cette méthode les tendances d'évolutions que nous voyons sur les deux premiers graphiques (méthodes RK et IF) sont corrigées sur le troisième (méthode *MIF0*), de manière à obtenir des quantités conservées qui ne possèdent quasiment aucune tendance. Cela nous permet de nous approcher d'une méthode réversible, ce qui est loin d'être le cas pour les méthodes *RK* et*IF*.

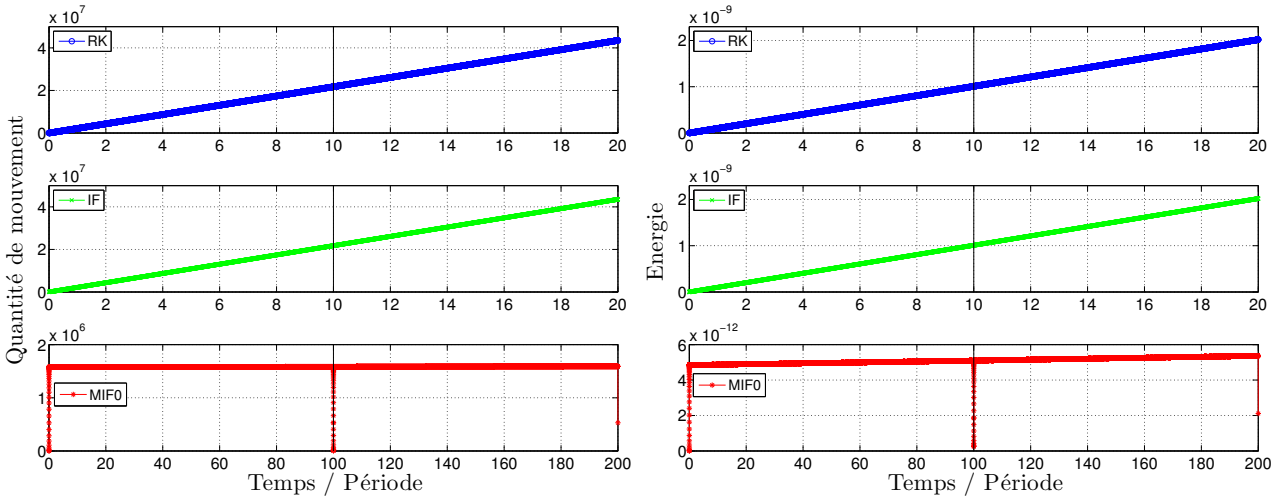

FIGURE VI.5 – Conservations de la quantité de mouvement (gauche) et de l'énergie (droite) pour un aller-retour de l'onde initiale sur 10 ou 100 périodes. Le tout est réalisé pour différents schémas numériques (*RK* en bleu, *IF* en vert et *MIF0* en rouge). Les paramètres sont : tolérance à 10^{-12}, $a = 0.1$ et $m = 0.6$.

Tableau des gains entre méthodes

Nous avons le tableau VI.2 récapitulant les facteurs de gains pour les différents jeux de paramètres a et m, avec une tolérance de 10^{-12} et 256 points. Nous ne mettons pas les gains de la méthode *IF* par rapport à la méthode *RK*, puisque toutes les deux donnent des résultats identiques.

TABLE VI.2 – Gains sur la valeur du pas de temps de calculs entre deux méthodes pour une même simulation. Ces gains de la méthode *MIF0* sont définis par rapport à la méthode *IF* classique.

m		0.9999	0.99	0.8	0.6	0.4	0.2
a=0.1	**MIF0**	11.31	12.02	13.49	15.04	17.76	25.84
a=0.2	**MIF0**	13.30	14.40	17.81	21.45	28.10	x
a=0.3	**MIF0**	14.92	16.69	21.76	27.34	x	x
a=0.4	**MIF0**	16.49	18.77	25.15	32.68	x	x

Nous voyons que nous sommes capables d'obtenir des gains très importants avec le facteur intégrant modifié par rapport au facteur intégrant classique. En effet, une même simulation peut aller jusqu'à près de 33 fois plus vite avec le facteur intégrant modifié. Une fois de plus, nous constatons que plus nous prenons un profil sinusoïdal pour l'onde et plus nous augmentons son amplitude, plus notre méthode est performante.
Si nous faisons de même pour une tolérance plus grande en 10^{-10}, ces gains seront environ 2 fois moins grands. Si nous prenons une tolérance encore plus grande en 10^{-8}, ces gains seront environ 4 fois moins grands. Cela s'explique par le fait qu'en diminuant la tolérance, les pas de temps vont être diminués d'une manière plus importante pour la méthode avec le facteur intégrant classique qu'avec celle du facteur intégrant modifié. Ainsi, en accumulant ces écarts en baissant progressivement la tolérance, nous obtenons des gains conséquents à de faibles tolérances.

3 Résultats numériques avec les autres méthodes

Dans cette section, pour les mêmes paramètres que précédemment, en VI.3.1 nous donnons les résultats numériques de simulations avec la méthode temporelle de Runge-Kutta de Verner et, en VI.3.2, nous présentons ceux pour le facteur intégrant généralisé à la place de notre facteur intégrant modifié.

3.1 Méthode temporelle de Verner

En utilisant la méthode d'avance temporelle d'ordre élevé de Verner, nous obtenons que le facteur intégrant modifié à l'ordre 0 fait jeu égal, sur la taille des pas de temps, avec le facteur intégrant classique et ne nous apporte, au mieux, que quelques pour cents de différences aux ordres 1 et 2. Nous pouvons le constater sur la figure VI.6 des pas de temps en fonction du

temps normalisé par la période, pour la simulation de paramètres $a = 0.6$, $m = 0.3$ et de tolérance 10^{-12}.

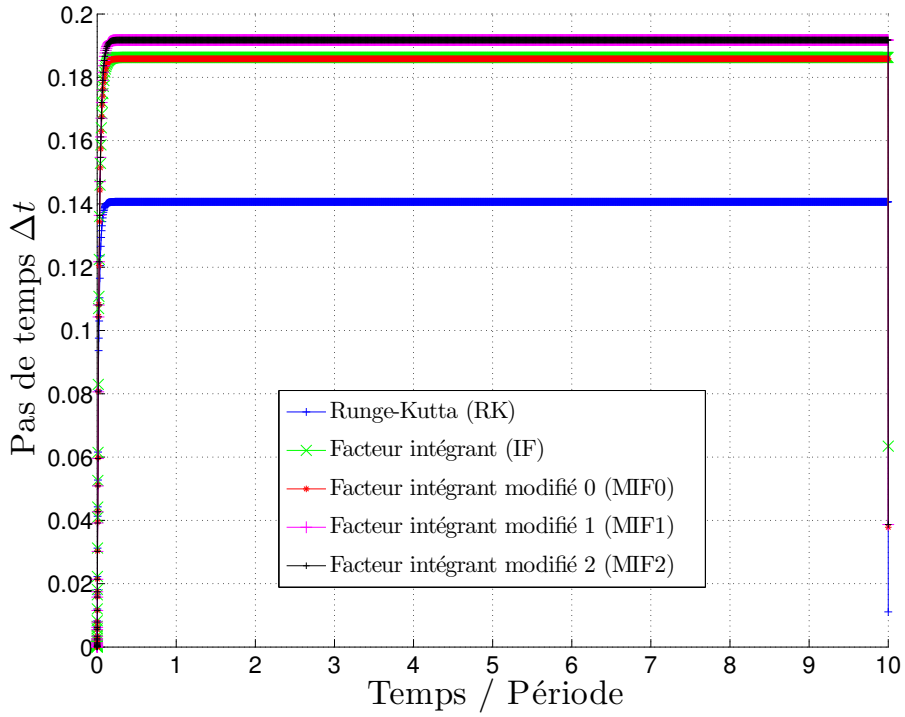

FIGURE VI.6 – Evolution du pas de temps Δt en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet). Les paramètres sont : tolérance à 10^{-12}, $a = 0.6$ et $m = 0.3$.

Nous pouvons en déduire que pour une avance temporelle d'ordre élevé, le facteur intégrant modifié ne semble pas être indispensable.

3.2 Adaptation du facteur intégrant généralisé

Le facteur intégrant généralisé de Krogstad (cf. III.4.2.b) donne des résultats similaires à notre méthode de facteur intégrant modifié dans tous les cas précédents.

Pour la méthode temporelle de bas ordre de Bogacki et Shampine, il est possible d'utiliser les ordres 1 et 2 du facteur intégrant généralisé, alors qu'à cause de la limite du Dense Output fourni, le facteur intégrant modifié ne peut dépasser l'ordre 0 (voir IV.1.4). L'ordre 1 du facteur intégrant généralisé n'apporte pas beaucoup plus de gains que le facteur intégrant modifié à l'ordre 0. Au mieux nous obtenons un facteur gain de 29.59 là où nous en avions un de 25.15 pour $m = 0.8$ et $a = 0.4$. Au pire nous faisons jeu égal pour $m = 0.8$ et $a = 0.1$ avec un gain de 13.49. Voire même, le facteur intégrant généralisé à l'ordre 1 peut faire moins bien que le facteur intégrant modifié à l'ordre 0 pour de très faibles amplitudes, comme nous le voyons sur la figure VI.7 représentant les gains entre méthodes en fonction de l'amplitude initiale, pour le paramètre cnoïdal $m = 0.8$ et une tolérance de 10^{-12}.

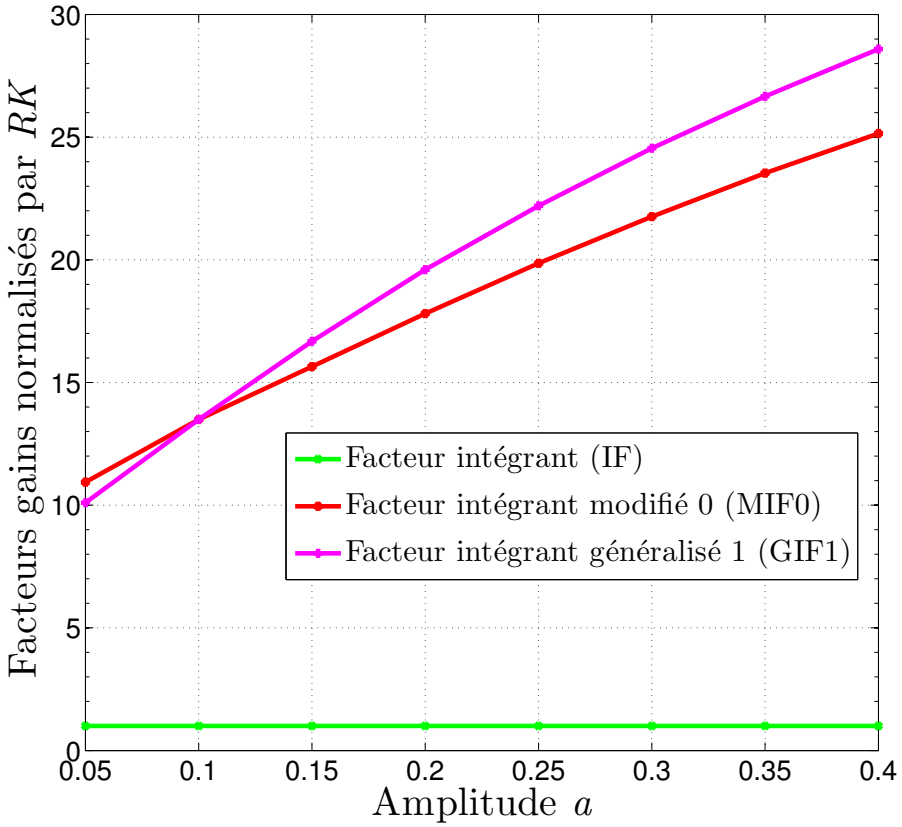

FIGURE VI.7 – Evolution du facteur gain entre différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge et *GIF1* en violet) en fonction de la tolérance. Les paramètres sont : tolérance à 10^{-12} et $m = 0.8$. Tous les gains sont calculés par rapport à la méthode *RK*.

Donc utiliser le facteur intégrant généralisé à l'ordre 1 ne semble pas indispensable. L'ordre 2 du facteur intégrant généralisé étant soumis à de fortes erreurs numériques, il n'est pas utilisable. Nous pouvons donc dire que ne pas obtenir ces ordres avec le facteur intégrant modifié n'est pas un problème.

4 Comparaison des erreurs numériques commises

Puisque nous connaissons la solution exacte de l'onde cnoïdale que nous propageons numériquement, nous pouvons regarder l'erreur numérique que nous commettons à chaque instant. Pour cela, nous traçons l'évolution du maximum de différences entre les valeurs de l'onde exacte et de l'onde simulée, pour les trois schémas temporels utilisés. Pour chacun d'entre eux, nous montrons les erreurs pour les méthodes de Runge-Kutta classique, de facteur intégrant classique et de facteur intégrant modifié (ou généralisé). Nous prenons le cas de l'onde cnoïdale définie par les paramètres $m = 0.8$ et $a = 0.3$, à la tolérance 10^{-12}, puisque pour ce cas, notre facteur intégrant modifié est plus intéressant que le facteur intégrant classique.

En utilisant l'avancement temporel de bas ordre de Bogacki et Shampine, nous voyons sur la figure VI.8 que, ces erreurs sur la surface libre, sont à l'avantage de notre facteur intégrant modifié à l'ordre 0 et du facteur intégrant généralisé à l'ordre 1, par rapport aux deux autres

méthodes. De plus, puisque cet écart est de plusieurs ordres de grandeurs, cela veut dire que nous obtenons à la fois les solutions 20 fois plus rapidement (voir le tableau VI.2), mais aussi de manière plus précise avec notre nouvelle méthode. Comme nous le voyons, pour notre facteur intégrant modifié nous obtenons ces erreurs oscillantes autour d'une valeur constante sur 10 périodes, alors que pour les simulations *RK* et *IF* nous ne pouvons pas obtenir les résultats sur ce même temps pour cause de temps de calcul trop important. Enfin, même si le facteur intégrant généralisé à l'ordre 1 nous donne un gain parfois plus important que pour notre facteur intégrant modifié à l'ordre 0, nous voyons ici que cela se fait au détriment des erreurs, puisque la méthode *GIF1* possède aussi des erreurs d'environ plus de 2 fois supérieures à la méthode *MIF0*.

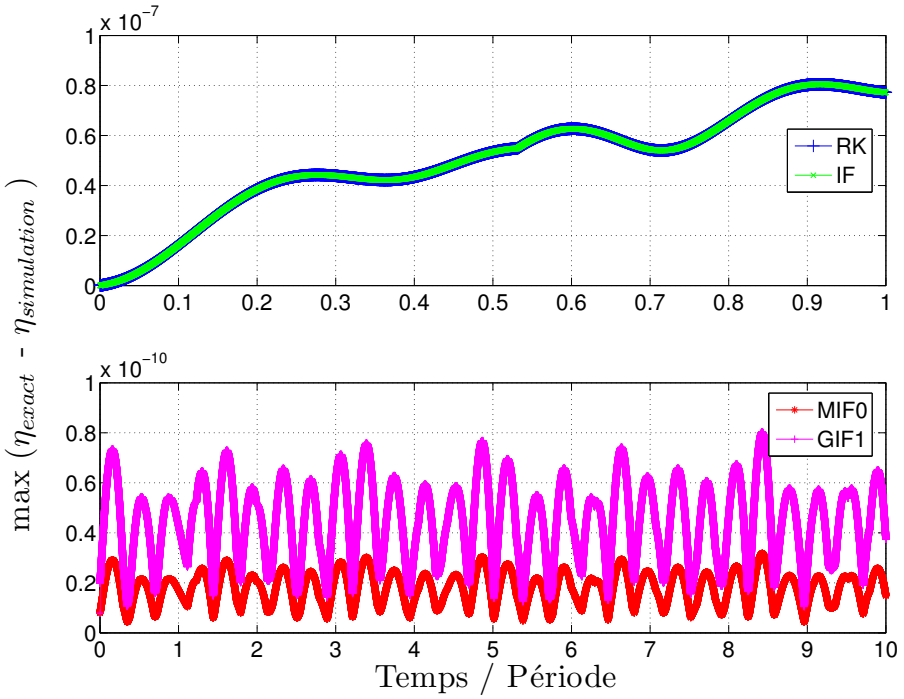

FIGURE VI.8 – Evolution du maximum de l'erreur entre l'onde exacte et l'onde simulée. Cas de l'avance temporelle de Bogacki et Shampine pour différentes méthodes : *RK* en bleu, *IF* en vert, *MIF0* en rouge et *GIF1* en violet. Tolérance à 10^{-12}, $m = 0.8$ et $a = 0.3$.

En ce qui concerne le schéma temporel de Dormand et Prince, sur la figure VI.9, il apparaît qu'au début de la simulation, le facteur intégrant modifié possède une erreur plus importante que les autres méthodes, d'un à deux ordres de grandeurs. Nous pouvons noter que cette erreur oscille autour d'une valeur moyenne, tandis que celles pour les méthodes de Runge-Kutta classique et de facteur intégrant classique ne cessent de croître (tout en oscillant aussi autour d'une valeur moyenne). A tel point, qu'au bout d'une vingtaine de périodes de simulations, figure VI.10, nous voyons que toutes les erreurs sont identiques, et ce, pour toutes les méthodes. En effet, pour les deux premières méthodes (*RK* et *IF*), l'erreur augmente jusqu'à atteindre la valeur moyenne de l'erreur de nos méthodes *MIF*. Une fois ces erreurs au même niveau, elles se mettent toutes à évoluer de la même manière et en même temps.

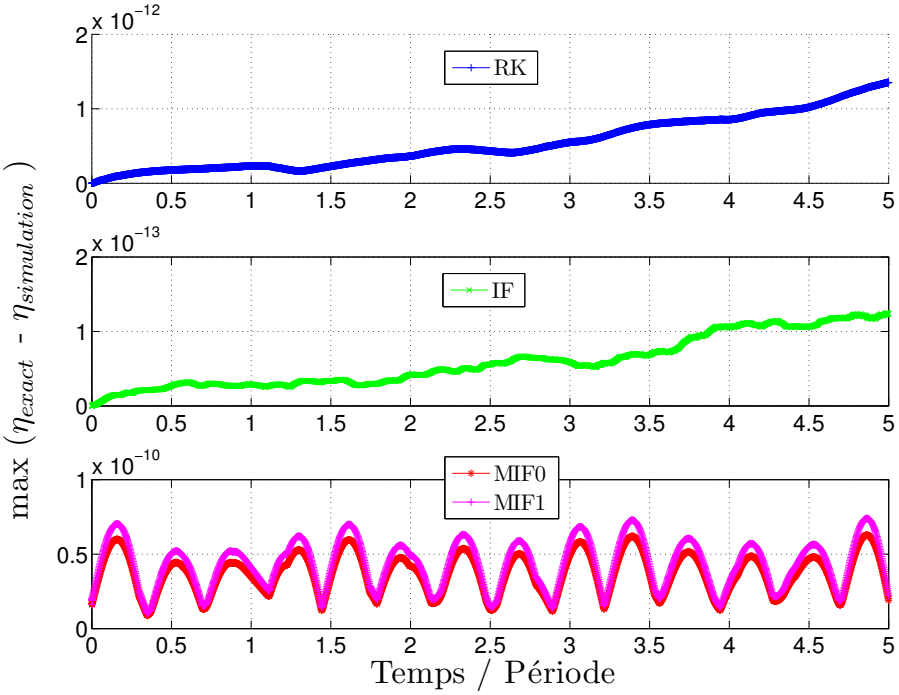

FIGURE VI.9 – Evolution du maximum de l'erreur entre l'onde exacte et l'onde simulée. Cas de l'avance temporelle de Dormand et Prince pour différentes méthodes : *RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet. Tolérance à 10^{-12}, $m = 0.8$ et $a = 0.3$. Simulations sur 5 périodes.

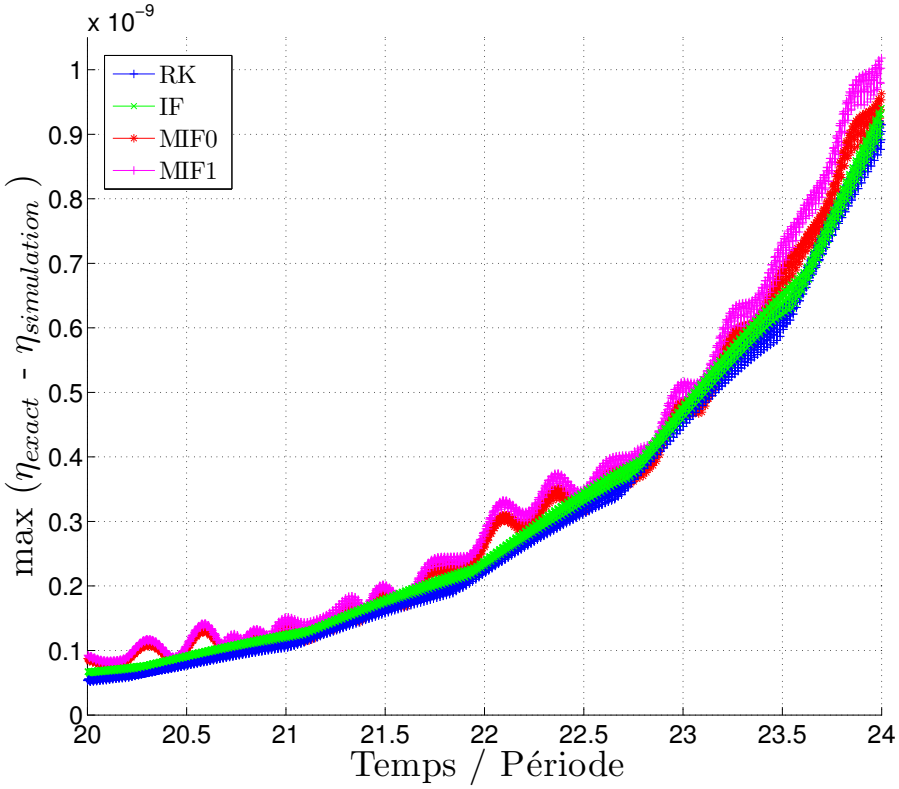

FIGURE VI.10 – Simulations identiques à la figure VI.9 pour les périodes 20 à 24.

Enfin, pour le schéma temporel d'ordre élevé de Verner, nous pouvons remarquer sur la figure VI.11 que les erreurs commises avec notre facteur intégrant modifié aux ordres 0 et 1 sont d'un ordre de grandeur plus important que pour les deux autres approches (*RK* et *IF*). Mais comme nous avons vu qu'avec ce schéma de Runge-Kutta, notre facteur intégrant modifié n'est pas intéressant en terme d'efficacité supplémentaire par rapport au facteur intégrant classique, obtenir ces erreurs légèrement plus grandes n'est pas un problème en soit.

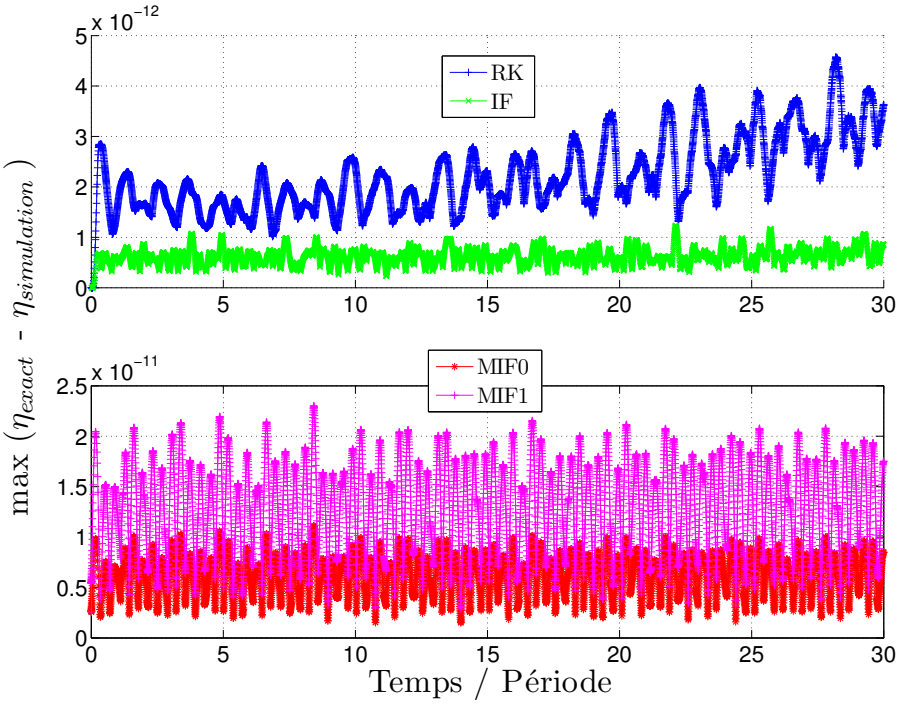

FIGURE VI.11 – Evolution du maximum de l'erreur entre l'onde exacte et l'onde simulée. Cas de l'avance temporelle de Verner pour différentes méthodes : *RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet. Tolérance à 10^{-12}, $m = 0.8$ et $a = 0.3$.

5 DISCUSSION

De ces simulations, il ressort ce que nous pensions, c'est-à-dire que nos intégrateurs exponentiels modifiés peuvent être très efficaces. Et ce, avec des pas de temps 33 fois plus grands que ceux que nous pouvions obtenir avec le facteur intégrant classique, pour de fortes amplitudes de l'onde considérée et tout en améliorant l'évolution des quantités conservées. De plus, entre les différentes approches avec ou sans facteur intégrant, les erreurs entre solutions exacte et simulée sont soit similaires, soit à notre avantage pour le schéma temporel de bas ordre. Cela nous montre que les gains que nous obtenons avec notre méthode ne se font pas au dépend de la Physique.

Afin de pouvoir conclure définitivement là-dessus, il ne nous reste plus qu'à vérifier ce qui se passe avec les équations les plus compliquées que nous puissions prendre, celles de la méthode dite *HOS*.

Chapitre VII

Application au modèle *High-Order Spectral*

Pour terminer notre étude, nous présentons ici les résultats des simulations obtenues pour le modèle hautement non linéaire *High-Order Spectral*, étudié en I.2.5. Dans la section VII.1, nous regardons la propagation d'une onde de Stokes sans aucune interaction ou perturbation, tandis que dans la section VII.2 nous faisons appel à une perturbation de la condition initiale de type *Benjamin-Feir*. Dans la dernière partie de ce chapitre en VII.3, nous avons les résultats pour les simulations avec une avance temporelle de Verner ainsi que l'adaptation du facteur intégrant généralisé pour les cas avec et sans instabilité. Ces résultats ont été obtenus à l'aide des équations de West et Watson et sont identiques pour les équations de Dommermuth et Yue.

1 Propagation sans perturbation de l'onde initiale

Cette section contient les résultats des simulations pour les méthodes temporelles de Dormand et Prince en VII.1.1 et de Bogacki et Shampine en VII.1.2.

1.1 Méthode temporelle de Dormand et Prince

1.1.a *Petites et moyennes cambrures*

Dans cette sous-section nous nous intéressons à des cambrures comprises entre 0.10 et 0.20. Pour ces simulations, nous avons réalisé une évolution temporelle d'une période d'onde de Stokes permanente sur 100 périodes temporelles. Le paramètre d'ordre M, qui fixe la troncature de

la décomposition en série de puissance de la surface libre η et du potentiel à la surface ϕ (voir l'annexe A), est pris égal à 4 et le nombre de points $N = 1024$.

Lois de conservations

Pour démontrer que nos nouveaux schémas numériques sont tout aussi corrects que les précédents en terme de conservation de quantités, prenons le cas de l'onde de Stokes de cambrure 0.20 à la tolérance 10^{-12}. Sur la figure VII.1 nous traçons les variations de la masse, de l'impulsion et de l'énergie pour les différentes méthodes. Il apparaît clairement que toutes ces quantités sont conservées de la même façon et ce quelque soit la technique utilisée. Au bout de 100 périodes de simulation, nous ne voyons aucune évolution dans la tendance des oscillations, qui restent bien centrées autour de la même valeur. Donc les méthodes avec et sans facteur intégrant sont toutes correctes les unes vis à vis des autres sur ce point.

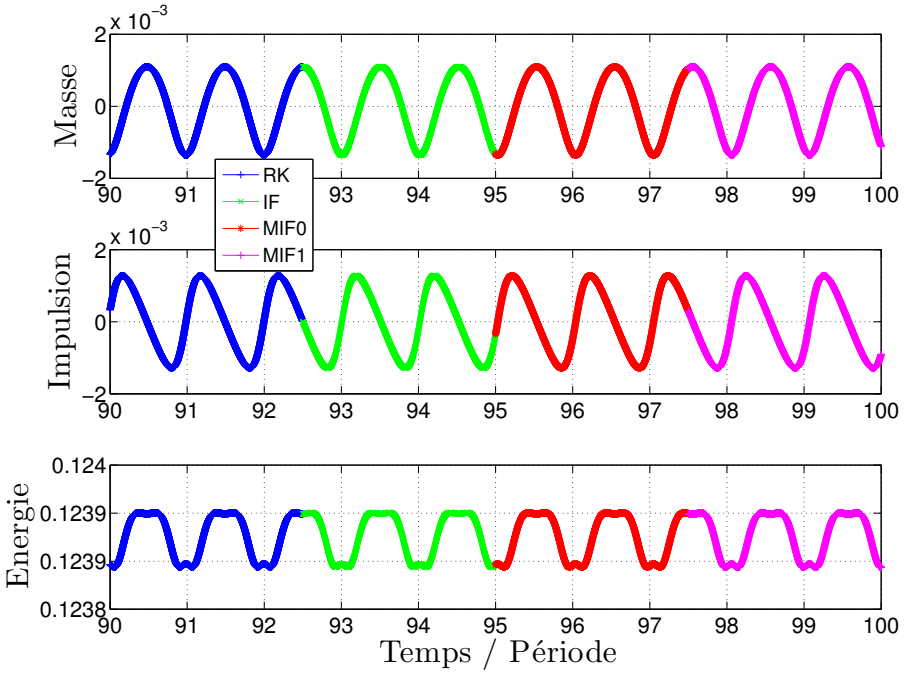

FIGURE VII.1 – Conservations de la masse, de l'impulsion et de l'énergie en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge et *MIF1* en violet). Les paramètres sont : tolérance à 10^{-12} et cambrure à 0.20.

Tableau des pourcentages de réduction entre méthodes

En faisant varier la cambrure et la tolérance, nous observons dans le tableau VII.1 que la méthode du facteur intégrant modifié à l'ordre 0, *MIF0*, n'apporte aucune réduction du nombre de boucles de calculs par rapport à la méthode du facteur intégrant classique, *IF*. Par contre, le facteur intégrant modifié à l'ordre 1, *MIF1*, permet d'obtenir une réduction entre 7 % et 23 % de pas de temps de calculs pour une même simulation par rapport au facteur intégrant classique. Nous constatons aussi que plus la cambrure augmente, plus le facteur intégrant modifié est intéressant alors que le facteur intégrant classique perd de son efficacité. En revanche, notre méthode à l'ordre 2, *MIF2*, apporte, certes, une réduction non négligeable par rapport au facteur intégrant classique mais inférieur à la méthode du facteur intégrant modifié d'ordre 1.

TABLE VII.1 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0*, *MIF1* et *MIF2* sont définies par rapport à la méthode *IF* classique.

Cambrure		0.10	0.13	0.20
tol=10^{-8}	**IF**	43.13	40.57	33.95
	MIF0	- 0.44	- 0.36	0.26
	MIF1	7.26	14.58	23.00
	MIF2	5.36	7.28	12.16
tol=10^{-10}	**IF**	43.22	40.66	34.03
	MIF0	- 0.44	- 0.37	0.26
	MIF1	7.20	14.60	23.03
	MIF2	5.12	7.08	12.05
tol=10^{-12}	**IF**	43.24	40.68	34.04
	MIF0	- 0.44	- 0.37	0.26
	MIF1	7.20	14.60	23.03
	MIF2	5.08	7.05	12.03

Enfin, il est à noter qu'avec des paramètres d'ordre $M = 6$ et $M = 8$ nous obtenons des résultats identiques, ce qui semble indiquer que pour ces cambrures, l'influence du choix de M n'a pas d'importance significative.

1.1.b *Fortes cambrures*

Dans cette section nous regardons des cambrures comprises entre 0.30 et 0.40 tout en variant la tolérance. Les paramètres des simulations sont les mêmes que précédemment, à la différence près que la fréquence de coupure k_m dans l'espace modal varie avec le profil de l'onde, comme expliqué en II.1.4.e.
A cause du solveur de la vitesse verticale W implémenté, nous avons une accumulation d'erreurs numériques très importantes, quelle que soit la méthode temporelle utilisée, ce qui fait que nous ne pouvons pas propager de telles ondes sur de longues périodes. Comme notre objectif n'est pas de développer un nouveau solveur spatial mais de nous intéresser au solveur temporel, nous allons comparer les résultats pour nos méthodes sur quelques périodes uniquement, celles durant lesquelles la solution se propage correctement, c'est-à-dire lorsque les erreurs numériques sont encore faibles.
Encore une fois, avec le tableau de valeurs VII.2, nous observons des pourcentages de réductions conséquents pour le facteur intégrant modifié à l'ordre 1 par rapport au facteur intégrant classique, réductions qui augmentent avec la cambrure, alors que dans le même temps le facteur intégrant classique perd encore de son efficacité par rapport à la méthode de Runge-Kutta classique. La méthode du facteur intégrant modifié d'ordre 0 est toujours sans intérêt et celle d'ordre 2 est encore un cran en-dessous de celle d'ordre 1.

TABLE VII.2 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0*, *MIF1* et *MIF2* sont définies par rapport à la méthode *IF* classique.

Cambrure		**0.30**	**0.35**	**0.40**
tol=10^{-8}	**IF**	24.36	19.03	13.05
	MIF0	0.29	- 0.22	- 0.88
	MIF1	21.09	22.77	26.89
	MIF2	17.39	20.01	23.99
tol=10^{-10}	**IF**	24.48	19.09	13.09
	MIF0	0.28	- 0.20	- 0.88
	MIF1	21.15	22.87	27.04
	MIF2	17.40	20.09	24.06
tol=10^{-12}	**IF**	24.52	19.11	13.11
	MIF0	0.28	- 0.20	- 0.88
	MIF1	21.18	22.92	27.08
	MIF2	17.41	20.12	24.08

Une fois de plus, avec des paramètres d'ordre $M = 6$ et $M = 8$ nous obtenons des résultats identiques.

1.2 Méthode temporelle de Bogacki et Shampine

Tout comme pour les équations de *Serre* en VI.2, c'est en faisant appel à une méthode de Runge-Kutta d'ordre peu élevé que nous obtenons des résultats beaucoup plus spectaculaires. En effet, avec ce schéma temporel, la méthode de facteur intégrant modifié peut nous permettre d'obtenir des pourcentages de réductions dépassant les 99 % sur le facteur intégrant classique, ce dernier ne faisant pas beaucoup mieux que l'approche de Runge-Kutta classique.
Comme expliqué en IV.3.2, à la place du pourcentage de réduction, nous donnons ici le facteur de gain (sur la valeur du pas de temps moyen ou du nombre total de pas de temps puisque ces deux valeurs sont identiques) entre les méthodes pour une meilleure visibilité des résultats.

Pas de temps Δt

Prenons par exemple le cas de l'onde de Stokes définie par la cambrure 0.10 et avec une tolérance de 10^{-4}. Nous voyons sur la figure VII.2 qu'il est tout à fait possible de réaliser une simulation aller-retour sur une centaine de périodes sans aucun problème de chute du pas de temps qui serait liée à une accumulation d'erreurs numériques. De plus, il apparaît que pour les méthodes de Runge-Kutta classique et du facteur intégrant classique, les pas de temps sont bien plus petits que pour la méthode du facteur intégrant modifié à l'ordre 0. Cela nous permet, ici, d'avoir un pas de temps 35 fois plus grand par rapport à la méthode du facteur intégrant

classique et donc d'être plus rapide avec ce même facteur en terme de nombre total de boucles de calcul.

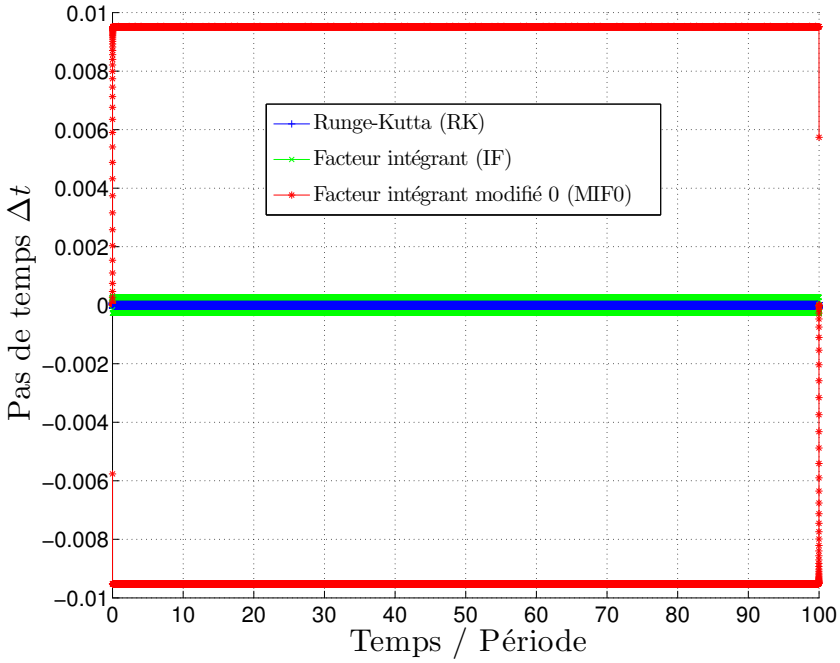

FIGURE VII.2 – Evolution du pas de temps Δt en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert et *MIF0* en rouge). Les paramètres sont : tolérance à 10^{-4} et cambrure à 0.10.

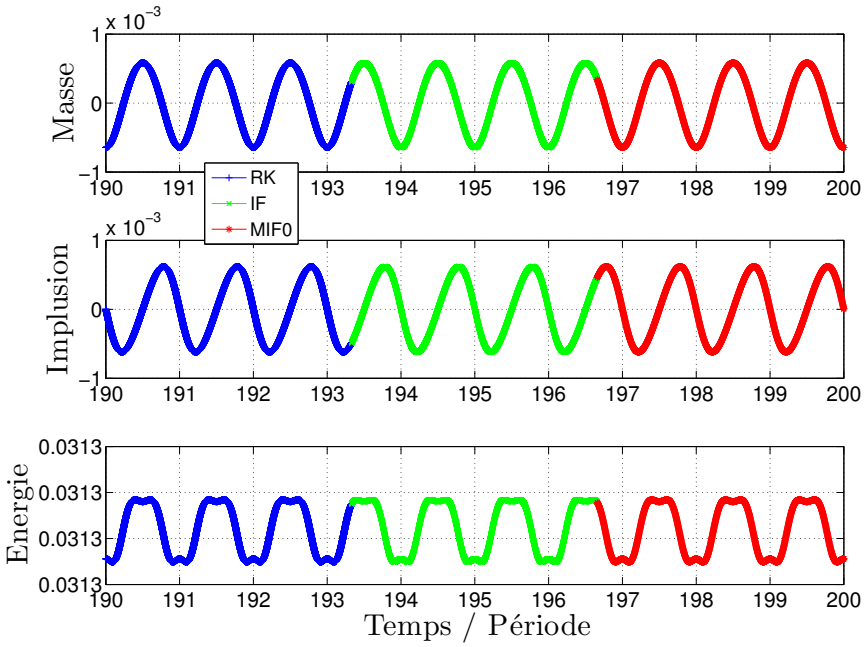

FIGURE VII.3 – Conservations de la masse, de l'impulsion et de l'énergie en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert et *MIF0* en rouge). Les paramètres sont : tolérance à 10^{-4} et cambrure à 0.10.

Lois de conservations

Pour être certain que nos méthodes permettent de calculer correctement la solution, nous pouvons tracer sur la figure VII.3 les 10 dernières périodes d'évolution de la masse, de l'impulsion et de l'énergie pour la simulation retour, après avoir fait un aller sur 100 périodes. Donc sur ce graphique nous voyons ces données pour les périodes 190 à 200. Il en ressort que les trois quantités qui doivent être conservées le sont bien, puisque pour toutes les méthodes nous ne voyons aucune tendance de déviation de ces quantités qui oscillent autour d'une même valeur moyenne. Cela s'explique par le fait que, pour obtenir une évolution dans cette plage de variation acceptable, selon la tolérance choisie, le pas de temps sera automatiquement pris plus ou moins grand.
Pour terminer de confirmer nos résultats, il a aussi été vérifié que les spectres de Fourier des deux variables $\hat{\eta}$ et $\hat{\phi}^s$ à la fin des simulations sont bien tous identiques entre les différentes méthodes.

Tableau des gains entre méthodes

Sur le tableau récapitulatif VII.3 pour lequel nous avons fait varier la cambrure de l'onde et la tolérance, il ressort encore une fois que plus la cambrure (et donc la difficulté de calcul) augmente, plus le facteur intégrant classique perd de son importance en passant d'un facteur gain 3 à 1.5 par rapport à la méthode Runge-Kutta classique. Par contre, dans le même temps, nous voyons que le facteur intégrant modifié à l'ordre 0 permet des gains de plus en plus importants. Au final, à l'aide de notre facteur intégrant modifié nous pouvons obtenir un pas de temps plus grand et donc l'obtention du résultat final en allant plus vite par un facteur compris entre 35 et 630 000, selon la tolérance choisie, par rapport au facteur intégrant classique.

Quand la cambrure est comprise entre 0.10 et 0.20, l'évolution du pas de temps est constante et nous pouvons faire propager notre onde de test sur des centaines ou milliers de périodes, aller et retour. Ces grandes simulations sont effectuées avec les tolérances les plus fortes : 10^{-4} et 10^{-6}. Pour des tolérances à partir de 10^{-8}, puisque le pas de temps doit obligatoirement être de plus en plus petit pour satisfaire le contrôle imposé de l'erreur dans le pas adaptatif (et ce, afin d'obtenir l'évolution des quantités en VII.3) nous ne pourrons pas aller si loin en temps, à cause de temps de calcul trop importants. En effet, pour une onde de cambrure 0.10 et une tolérance de 10^{-4} les pas de temps ont pour ordre de grandeur 10^{-4} avec les méthodes de Runge-Kutta classique et du facteur intégrant classique, contre 10^{-2} avec le nouveau facteur intégrant modifié à l'ordre 0. Pour une tolérance de 10^{-12}, nous avons pour les pas de temps des ordres de grandeurs de 10^{-12} avec les méthodes de Runge-Kutta classique et du facteur intégrant classique, contre 10^{-6} avec le nouveau facteur intégrant modifié à l'ordre 0. Ainsi, en diminuant la tolérance d'un facteur 100, l'évolution des pas de temps passe aussi par une réduction d'un facteur 100 pour les méthodes de Runge-Kutta classique et du facteur intégrant classique, alors que dans le même temps le facteur intégrant modifié ne diminue le pas de temps que d'un facteur 10. Ceci explique les gains que nous obtenons et que les simulations sur des temps longs à de si faibles tolérances ne peuvent devenir envisageables qu'avec notre facteur intégrant modifié. Néanmoins, à ces petites tolérances, nous pouvons tout de même vérifier sur un temps très court que pour toutes les simulations nous retrouvons bien des évolutions identiques, tout comme en VII.3.

Pour les cas des très fortes cambrures, comprises entre 0.30 et 0.40, nous ne pouvons pas propager nos ondes sur plus de quelques périodes comme expliqué précédemment.

TABLE VII.3 – Gains sur la valeur du pas de temps de calculs entre deux méthodes pour une même simulation. Les gains *IF* sont définis entre les méthodes *IF* et *RK*, tandis que les gains *MIF0* sont définis par rapport à la méthode *IF* classique.

Cambrure		**0.10**	**0.13**	**0.20**	**0.30**	**0.35**	**0.40**
tol $= 10^{-4}$	**IF**	2.89	2.59	2.18	1.83	1.68	1.50
	MIF0	35.40	38.49	44.77	53.03	57.02	57.42
tol $= 10^{-6}$	**IF**	2.89	2.59	2.18	1.83	1.68	1.50
	MIF0	354.01	384.94	447.97	527.54	570.13	622.82
tol $= 10^{-8}$	**IF**	2.89	2.59	2.18	1.83	1.68	1.50
	MIF0	3 540.29	3 850.01	4 481.93	5 276.77	5 707.06	6 251.11
tol $= 10^{-10}$	**IF**	2.89	2.59	2.18	1.83	1.68	1.50
	MIF0	35 402.87	38 500.14	44 819.63	52 767.73	57 071.12	62 513.43
tol $= 10^{-12}$	**IF**	2.89	2.59	2.18	1.83	1.68	1.50
	MIF0	354 028.80	385 001.54	448 196.29	527 677.26	570 711.30	625 134.69

Nos résultats sont alors obtenus sur la partie des simulations n'ayant pas encore été soumise aux erreurs numériques. Aussi, le pas de temps n'est pas constant mais évolue un peu en oscillant. Donc, pour les calculs de gains entre les méthodes sur la taille du pas de temps nous prenons la valeur du pas de temps moyen sur la simulation. Ces données sont présentées pour montrer ce qui se dessine depuis plusieurs chapitres, à savoir que nos gains augmentent avec la difficulté de calcul, qui est ici directement liée à la cambrure.

2 Instabilité modulationnelle de Benjamin et Feir

En 1967, Benjamin et Feir ont observé expérimentalement [4] qu'à l'aide d'un batteur situé à une extrémité d'un bassin, générant un train d'ondes périodiques, il était possible de faire apparaître à sa surface des modulations d'amplitude suite à la présence d'une instabilité. Cette modulation est due aux échanges d'énergie entre le mode fondamental du spectre et les modes voisins, appelés satellites. Cette instabilité modulationnelle progressive d'un train d'onde de vagues, qui porte désormais le nom de ses inventeurs, a aussi été décrite par d'autres [53, 83]. Sous l'effet de cette instabilité, le train de vagues présente un cycle de modulation et de démodulation, appelé récurrence de Fermi, Pasta et Ulam [32]. Il a été suggéré que des vagues dites *scélérates* peuvent être créées au maximum de cette récurrence [29, 43, 61].

Après avoir précisé la perturbation utilisée ici, nous présentons les résultats des simulations pour les méthodes temporelles de Dormand et Prince en VII.2.1 et de Bogacki et Shampine en VII.2.2.

Perturbation de l'onde initiale

Dans le cas de ces instabilités obtenues en prenant un train d'onde non linéaire auquel nous ajoutons une modulation périodique de l'amplitude, parmi les profils d'ondes que nous étudions, seules les ondes définies par une cambrure inférieure à 0.20 sont préservées des erreurs numériques trop importantes et peuvent donc se propager avec le solveur utilisé. Précédemment nous faisions propager une seule période spatiale de ces ondes. Ici, nous allons faire varier le nombre de périodes entre 8 et 9 par boîte de calcul, de manière à obtenir uniquement deux modes perturbables dans le système, tels que le deuxième soit le plus excitable. Nous fixons le pourcentage de perturbation $\epsilon = 0.10$ introduit sur le profil initial des ondes de stokes. Il est à noter que les résultats suivants sont quasiment identiques si nous prenons $\epsilon = 0.05$ ou $\epsilon = 0.15$.

2.1 Méthode temporelle de Dormand et Prince

Pour les ondes de cambrure 0.10 nous prenons 9 périodes d'onde, avec une fréquence de coupure au mode 72 et 1440 points. Pour les ondes de cambrure 0.13 nous prenons 8 périodes, une fréquence de coupure pour le mode 64 et 1280 points. Enfin, nous faisons une simulation sur 500 périodes temporelles.

Après avoir fait des simulations en variant cambrure et tolérance, sur le tableau récapitulatif VII.4 nous pouvons constater qu'encore une fois nous obtenons des pourcentages de réduction du nombre de pas de temps de l'ordre de 10 % à 15 % avec le facteur intégrant modifié sur le facteur intégrant classique. En ce qui concerne le facteur intégrant modifié à l'ordre 2, si les

pourcentages de réductions sont supérieurs à la méthode d'ordre 1 uniquement pour l'onde de cambrure 0.13, cela est dû au fait qu'avec ce profil d'onde, la simulation est déjà fortement perturbée par des erreurs numériques. Ainsi, les erreurs numériques induites par l'ordre 2 du facteur intégrant modifié sont comme *noyées* dans les erreurs déjà présentes et ne se font pas *trop ressentir* sur le résultat.

TABLE VII.4 – Pourcentages de réduction du nombre total de boucles de calculs entre deux méthodes pour une même simulation. Les réductions *IF* sont définies entre les méthodes *IF* et *RK*, tandis que les réductions *MIF0*, *MIF1* et *MIF2* sont définies par rapport à la méthode *IF* classique.

Cambrure		**0.10**	**0.13**	**Cambrure**		**0.10**	**0.13**
tol=10^{-4}	**IF**	38.69	36.44	**tol**=10^{-10}	**IF**	43.39	38.32
	MIF0	0.05	-0.70		**MIF0**	0.35	0.48
	MIF1	14.91	12.03		**MIF1**	9.86	10.71
	MIF2	-6.06	15.44		**MIF2**	6.82	15.00
tol=10^{-6}	**IF**	42.18	37.87	**tol**=10^{-12}	**IF**	43.42	38.34
	MIF0	0.24	0.21		**MIF0**	0.35	0.49
	MIF1	10.79	10.88		**MIF1**	9.84	10.71
	MIF2	9.82	16.23		**MIF2**	6.74	14.97
tol=10^{-8}	**IF**	43.20	38.26				
	MIF0	0.33	0.43				
	MIF1	10.01	10.73				
	MIF2	7.31	15.20				

Avec des paramètres d'ordre $M = 6$ et $M = 8$ nous obtenons des résultats identiques.

2.2 Méthode temporelle de Bogacki et Shampine

Comme précédemment, nous utilisons maintenant l'avance temporelle de bas ordre de Bogacki et Shampine.

Pas de temps Δt

Les gains que nous obtenons ne se font pas au détriment de la physique. Nous pouvons le vérifier à la fois sur la figure des quantités conservées mais aussi sur l'évolution des pas de temps sur la figure VII.4 (normalisés par la valeur maximale et décalés pour éviter une superposition graphique). En effet, sur cette dernière nous voyons apparaître des cycles de récurrence pour toutes les méthodes utilisées, comme expliqué en introduction de cette section. Lorsque l'amplitude de l'onde augmente suite à l'instabilité, le calcul numérique devient plus difficile.
Une conséquence directe est que le pas de temps adaptatif décroît. Ainsi, cela nous montre que

les difficultés de calcul, dues à une évolution plus compliquée de l'onde, ont bien lieu au même instant et ce, quelque soit la méthode implémentée.

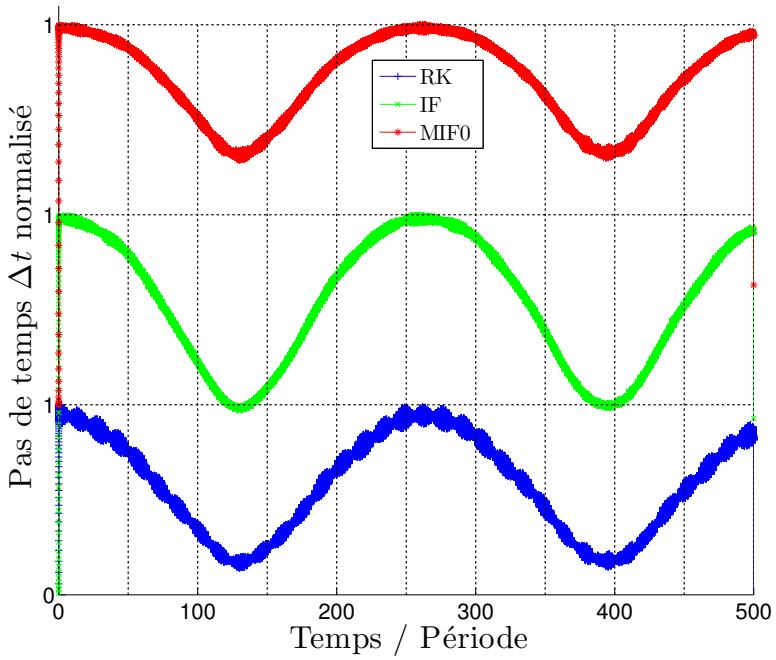

FIGURE VII.4 – Evolution du pas de temps Δt (normalisé par son maximum) en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert et *MIF0* en rouge). Les paramètres sont : tolérance à 10^{-4} et cambrure à 0.10.

Lois de conservations

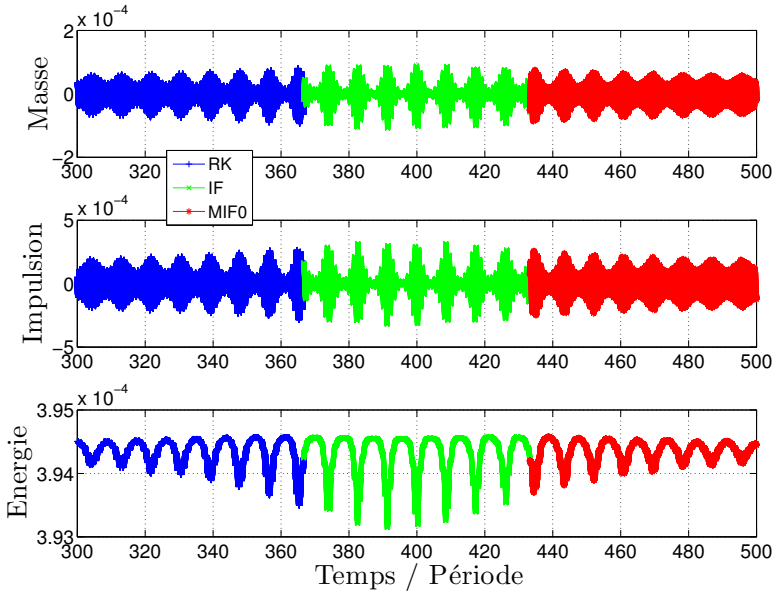

FIGURE VII.5 – Conservations de la masse, de l'impulsion et de l'énergie en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert et *MIF0* en rouge). Les paramètres sont : tolérance à 10^{-4} et cambrure à 0.10

Encore une fois, sur la figure VII.5 nous avons les différentes quantités conservées pour chacune des méthodes utilisées. Il en ressort que ces quantités évoluent toutes de la même manière et avec le même ordre de grandeur. Les petites variations que nous voyons sont normales, car liées à l'instabilité qui se manifeste entre les périodes temporelles 340 et 460.

Tableau des gains entre méthodes

Nous faisons varier la tolérance pour les cambrures qui ont été testées et propagées par le solveur *HOS* implémenté et nous notons les différents facteurs de gains dans le tableau VII.5.

TABLE VII.5 – Gains sur la valeur du pas de temps de calcul entre deux méthodes pour une même simulation. Les gains *IF* sont définis entre les méthodes *IF* et *RK*, tandis que les gains *MIF0* sont définis par rapport à la méthode *IF* classique.

Cambrure		**0.10**	**0.13**	**Cambrure**		**0.10**	**0.13**
tol=10^{-4}	**IF**	4.24	2.98	**tol**=10^{-10}	**IF**	4.38	3.52
	MIF0	26.27	37.28		**MIF0**	23 080.78	28 146.02
tol=10^{-6}	**IF**	4.39	3.53	**tol**=10^{-12}	**IF**	4.38	3.52
	MIF0	230.24	280.09		**MIF0**	230 803.73	281 452.06
tol=10^{-8}	**IF**	4.38	3.53				
	MIF0	2 313.26	2 826.40				

Les résultats sont presque tous issus de simulations d'une durée inférieure à quelques périodes temporelles (de la demi-période au centième de période) pour des raisons de temps de calculs très importants. Par contre, pour la tolérance la plus forte en 10^{-4}, cela a été fait pour 500 périodes, comme étudié dans la sous-section précédente.

Dans cette situation d'instabilité, nous avons toujours des gains très intéressants avec le facteur intégrant modifié, comme nous le voyons sur le tableau VII.5, avec des pas de temps plus grands d'un facteur compris entre 26 et 280 000 par rapport au facteur intégrant classique. Les écarts créés entre les méthodes selon la tolérance viennent toujours du fait qu'en diminuant la tolérance d'un facteur 100, les méthodes de Runge-Kutta classique et du facteur intégrant classique voient leurs pas de temps se réduire d'un facteur 100, alors que dans le même temps la méthode du facteur intégrant modifié n'a un pas de temps réduit que d'un facteur 10.
Enfin, nous voyons encore que lorsque le facteur intégrant classique perd de son efficacité sur la méthode de Runge-Kutta classique, passant par exemple d'un facteur 4.38 à 3.53 à la tolérance 10^{-6}, notre facteur intégrant modifié est plus intéressant, passant d'un facteur gain 230 à 280.

Nous pouvons constater, qu'ici, les gains du facteur intégrant classique sont plus grands environ d'un rapport 1.5 que pour le cas sans instabilité en VII.3. La conséquence directe de cette amélioration est que les gains du facteur intégrant modifié sont ici plus petits de ce même facteur que pour le cas sans instabilité. Nous retrouvons une remarque déjà formulée pour d'autres équations modèles, à savoir, que notre facteur intégrant modifié est plus intéressant lorsque le facteur intégrant classique perd de son intérêt (et vice versa).

3 Résultats numériques avec les autres méthodes

Dans cette section, que ce soit pour les cas sans et avec instabilité de Benjamin-Feir, en VII.3.1 nous donnons les résultats numériques des simulations faisant appel à la méthode temporelle de Runge-Kutta de Verner, puis, en VII.3.2, nous présentons les résultats avec le facteur intégrant généralisé à la place de notre facteur intégrant modifié. Les simulations ont été réalisées pour les mêmes paramètres que précédemment.

3.1 Méthode temporelle de Verner

En utilisant la méthode d'avance temporelle d'ordre élevé de Verner, le facteur intégrant modifié aux ordres 0, 1 et 2 n'apporte rien d'intéressant par rapport au facteur intégrant classique. Nous pouvons le constater sur la figure VII.6 des pas de temps en fonction du temps normalisé par la période. Les pas de temps des méthodes avec facteur intégrant sont tous identiques. Autant pour le cas d'une propagation sans perturbation (à gauche, cambrure 0.20) que pour le cas de l'instabilité de Benjamin-Feir (à droite, cambrure 0.10) à la tolérance 10^{-8}. De ce fait, pour la méthode temporelle d'ordre élevé, le facteur intégrant modifié n'est pas utile.

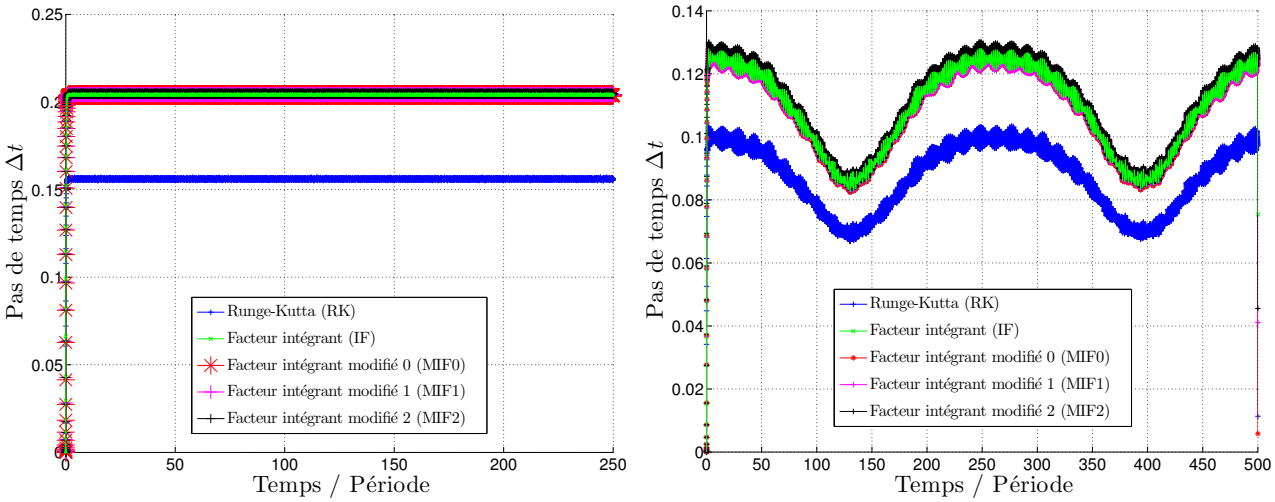

FIGURE VII.6 – Evolution du pas de temps Δt en fonction du temps de simulation normalisé par la période pour différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge, *MIF1* en violet et *MIF2* en noir). Cambrure 0.20 sans perturbation initiale à gauche et cambrure 0.10 avec instabilité de Benjamin-Feir à droite, pour une tolérance de 10^{-8}.

3.2 Adaptation du facteur intégrant généralisé

Le facteur intégrant généralisé de Krogstad donne des résultats similaires à notre méthode de facteur intégrant modifié dans tous les cas précédents.

Pour le cas de la méthode temporelle de bas ordre de Runge-Kutta de Bogacki et Shampine, pour laquelle notre facteur intégrant modifié ne peut dépasser l'ordre 0 (voir IV.1.4), nous pouvons utiliser le facteur intégrant généralisé aux ordres 1 et 2 (voir III.4.2.b).

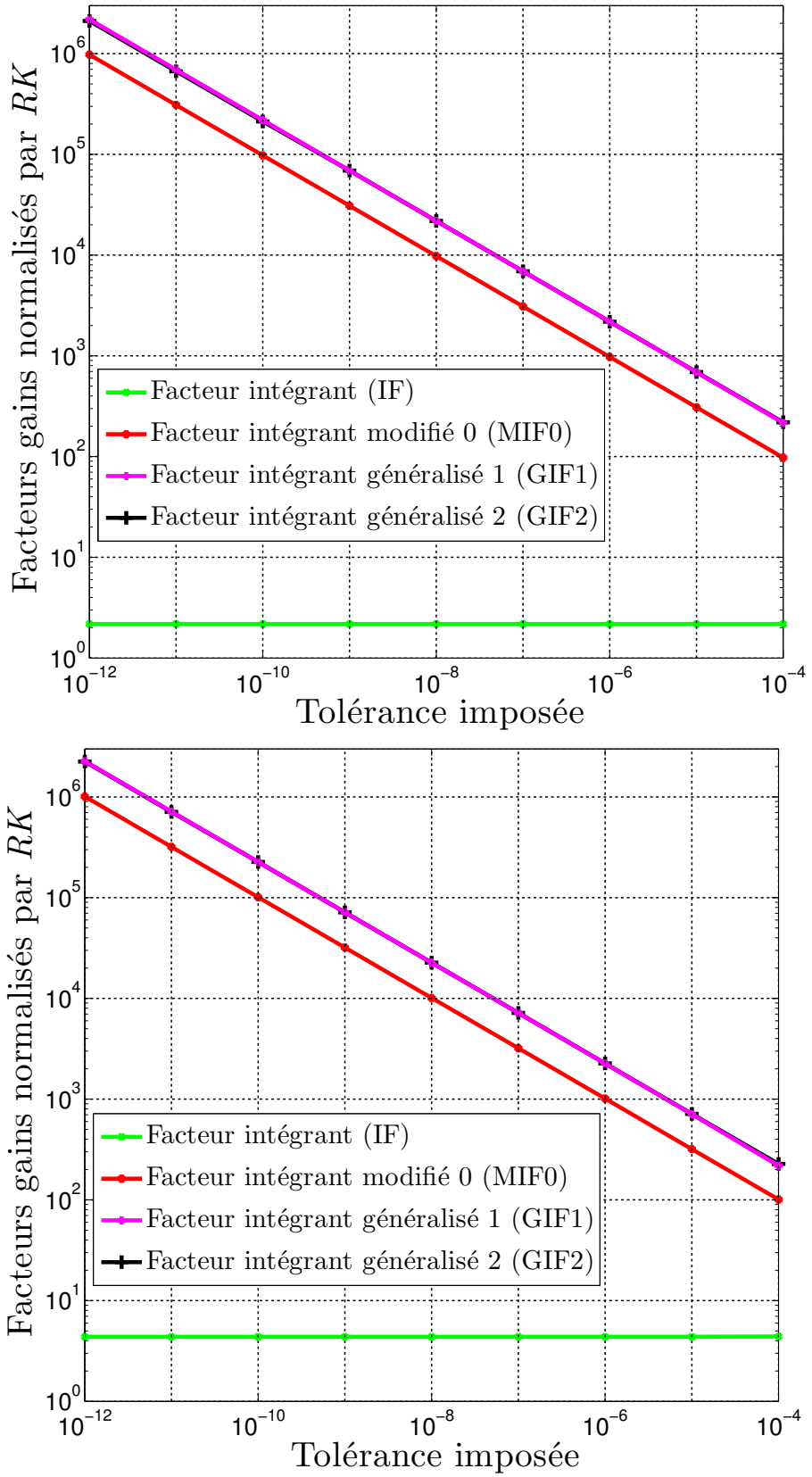

FIGURE VII.7 – Evolution du facteur gain entre différents schémas numériques (*RK* en bleu, *IF* en vert, *MIF0* en rouge, *GIF1* en violet et *GIF2* en noir) en fonction de la tolérance. Tous les gains sont calculés par rapport à la méthode *RK*. Cambrure 0.20 sans perturbation initiale en haut et cambrure 0.10 avec instabilité de Benjamin-Feir en bas.

Sur la figure VII.7, nous représentons les gains précédents en fonction de la tolérance, en haut pour la propagation sans perturbation d'une onde de cambrure 0.20 et en bas pour l'étude d'une instabilité de Benjamin-Feir avec une cambrure de 0.10. Nous comparons les gains pour les schémas de Runge-Kutta classique, de facteur intégrant classique, de facteur intégrant modifié à l'ordre 0, noté *MIF0*, et de facteur intégrant généralisé aux ordres 1 et 2, noté *GIF1* et *GIF2*. Pour une meilleure lisibilité graphique, nous avons calculé tous les gains des méthodes avec un facteur intégrant par rapport à la méthode de Runge-Kutta classique (cela équivaut à multiplier les gains des cases *MIF0* des tableaux VII.3 et VII.5 par ceux des cases *IF*). Les progressions linéaires des gains des méthodes *MIF0*, *GIF1* et *GIF2* s'expliquent par ce que nous avons déjà dit, à savoir, que les méthodes de Runge-Kutta classique et du facteur intégrant classique voient leurs pas de temps décroître d'un facteur 100 lorsque nous diminuons la tolérance d'un facteur 100, alors que dans le même temps, avec le facteur intégrant modifié (ou généralisé), le pas de temps n'est réduit que d'un facteur 10.

Comme nous pouvons le constater sur les tableaux récapitulatifs, il n'y a pas de différence majeure entre les cas sans et avec instabilité.

En faisant appel au facteur intégrant généralisé à l'ordre 1 (*GIF1*), nous obtenons des gains 2.2 fois plus importants que ceux du facteur intégrant modifié à l'ordre 0 (*MIF0*). Cela se fait tout en conservant aussi bien les quantités qui doivent l'être, comme sur la figure VII.3. Par contre, le facteur intégrant généralisé à l'ordre 2 (*GIF2*) n'apporte rien de plus que celui d'ordre 1. Nous voyons donc qu'il semblerait intéressant de pouvoir utiliser notre facteur intégrant modifié à l'ordre 1.

4 Discussion

Nous avons pu voir qu'en faisant appel aux équations les plus complexes à notre disposition pour la simulation de vagues, nous pouvons augmenter la taille du pas de temps de calcul avec notre facteur intégrant modifié, et donc accélérer la simulation en diminuant le nombre total de pas de calcul nécessaires, par un facteur compris entre 35 et 625 000 sur le facteur intégrant classique selon la tolérance choisie. Cela tout en conservant l'énergie, la masse et la quantité de mouvement du système.

Ces conservations sont de plus obtenues sans aucune tendance de déviation dans le temps et dans le même ordre de grandeur quelque soit la méthode utilisée.

Habituellement, il est pris une tolérance de l'ordre de 10^{-4} à 10^{-6}, ce qui nous donne une plage de gains sur le nombre total de boucles de calcul entre 35 et 623 avec l'utilisation du facteur intégrant modifié.

De plus, à l'aide de ce dernier il devient possible de faire des simulations à de très faibles tolérances pour un temps de calcul raisonnable, ce qui n'est tout simplement pas réalisable avec le facteur intégrant classique.

Cela peut servir, par exemple, pour vérifier des conditions locales de manière très précise et beaucoup plus rapide.

Conclusion

Comme expliqué dans le chapitre II, les simulations numériques de cette thèse ont été réalisées dans l'espace spectral de Fourier. Ce choix nous permet de reécrire les équations d'évolutions de manière plus simple d'un point de vue d'implémentation dans le code informatique, puisque, par exemple, le calcul des dérivées spatiales se fait à l'aide d'une simple multiplication par un multiple du nombre d'onde. De plus, à l'aide des *FFT*, nous pouvons passer de l'espace physique à l'espace de Fourier à tout moment et pour un temps de calcul raisonnable, ce qui nous permet de calculer les non linéarités dans l'espace le plus adapté puis de revenir dans l'autre afin de continuer l'évolution temporelle. Par contre, cette facilité d'utilisation a un prix, puisque en travaillant avec une méthode pseudo-spectrale nous devons nous *méfier* continuellement des erreurs numériques de repliement (*l'aliasing*), dues à la propagation d'erreurs des hautes fréquences vers les plus basses. Si nous ne prenons pas les bonnes mesures de prévention, ces erreurs peuvent croître à une vitesse telle que les simulations ne peuvent être réalisables. Il faut donc faire attention dès que nous calculons une non linéarité, pour éviter de propager ces erreurs. Une fois que nous maîtrisons cette technique, les méthodes pseudo-spectrales sont très efficaces. Elles nous permettent de simuler les équations des vagues (chapitre I) de manière très précise et sur des temps très longs.

Selon le cas étudié (équation, profil de l'onde, paramètres ...), le schéma numérique d'avancement temporel peut montrer ses limites en n'arrivant pas à manipuler de manière efficace les termes compliqués que sont les calculs des non linéarités. Pour dépasser ce problème nous faisons appel à une méthode de type implicite, ce qui a pour conséquence directe d'afficher un coût de calcul nettement supérieur aux méthodes explicites qui sont très utilisées. Si nous souhaitons garder notre méthode explicite favorite, souvent de type Runge-Kutta, il nous faut trouver une autre technique. Puisqu'une partie de la *raideur numérique* est issue du terme linéaire des équations, il a été développé de nombreuses méthodes dans le cadre des *intégrateurs exponentiels*, qui permettent de retirer ce terme afin de le calculer analytiquement. Quant à la partie non linéaire, elle est calculée par le schéma numérique. C'est le principe du facteur intégrant classique (chapitre III). Comme nous l'avons vu dans nos résultats de simulations, avec ce dernier nous sommes quasiment toujours capables d'augmenter la taille des pas de temps d'évolution par rapport à la méthode de Runge-Kutta seule, et donc de réduire le nombre total de calculs nécessaires à une même simulation. Mais il est possible d'aller plus loin et d'envisager de retirer plus de difficulté de calcul dans le système, en s'attaquant aussi à la partie

non linéaire. C'est ce que nous proposons à l'aide de notre facteur intégrant modifié (chapitre IV). En utilisant une approximation polynomiale du terme non linéaire, et en la retranchant à l'équation d'évolution, cela a pour conséquence de, parfois, permettre l'utilisation d'un pas de temps encore plus grand. La définition que nous avons choisie de cette approximation, à l'aide d'un développement polynomial de Taylor autour du temps initial d'une boucle de calcul, n'est peut être pas la plus optimale, mais elle nous permet de ne pas ajouter de calculs supplémentaires. En effet, nous réutilisons les calculs internes de l'avancement temporel de Runge-Kutta afin de construire notre polynôme aux différents ordres, en utilisant la technique du *Dense Output*, pour obtenir les dérivées de la solution qui nous sont nécessaires. Cette méthode est donc *gratuite* en terme de temps de calcul. Plus précisément, lorsque nous faisons appel à cette méthode pour les équations les plus *faciles* à simuler, comme celles de *Korteweg et de Vries*, de *Benjamin, Bona et Mahony* et de *Schrödinger Non Linéaire* (chapitre V), il existe un coût de manipulation numérique des données qui est non négligeable. Par contre, dès que nous faisons des simulations pour des équations modèles plus difficiles à manipuler, comme pour les équations de *Serre* (chapitre VI) et *High-Order Spectral* (chapitre VII), ce surcoût est bien négligeable.

Il s'avère que notre facteur intégrant modifié ressemble au facteur intégrant généralisé développé par Krogstad. A la place d'utiliser le Dense Output pour calculer les dérivées du polynôme au temps actuel, il a fait le choix d'utiliser des différences finies sur les temps précédents. Nous voyons donc une différence importante d'un point de vue numérique. Cette méthode est *multistep*, puisqu'elle fait appel à plusieurs données de temps différents, alors que la nôtre est *one step*, puisque nous ne regardons que le temps présent. C'est-à-dire que nous avons besoin de moins d'espace de stockage mémoire, ce qui peut être un atout, par exemple pour des problèmes en 3D.

Notre facteur intégrant modifié est conçu pour être totalement indépendant de la méthode de Runge-Kutta utilisée. Comme nous avons pu le voir tout au long de ce manuscrit, pour une méthode de Runge-Kutta d'ordre élevé (Verner, ordre 9), le facteur intégrant modifié ne nous est d'aucune utilité, puisqu'il ne fait ni mieux, ni pire, que le facteur intégrant classique en terme de largeur de pas de temps. Pour un schéma d'ordre moyen (Dormand et Prince, ordre 5), nous avons pu constater que selon le profil de l'onde et donc la complexité de calcul, nous pouvions réduire le nombre de pas de temps total d'environ 30 %. Par contre, le cas le plus intéressant reste celui de la méthode de Runge-Kutta de bas ordre (Bogacki et Shampine, ordre 3) pour laquelle nous obtenons des pas de temps 30 à 600 000 fois plus grands qu'avec le facteur intégrant classique et ce, pour le modèle le plus non linéaire à notre disposition, le modèle *HOS*. Si nous regardons les gains sur la plage des tolérances usuelles, nous pouvons réduire le nombre total de pas de temps d'un facteur 35 à 600. Mais le plus important est que nous n'obtenons pas ces résultats au dépend de la Physique. En effet, sur les différentes courbes représentant l'évolution des quantités qui doivent être conservées, nous nous apercevons que, pour notre facteur intégrant modifié, soit ces quantités évoluent de la même manière que pour les autres méthodes, soit nous arrivons à supprimer les tendances d'évolutions qui ne devraient pas exister. De plus, pour le cas des équations de *Serre*, nous avons pu vérifier que les erreurs entre solutions exacte et simulée sont tout à fait correctes ou à notre avantage.

Le dernier point à relever est que, plus nous augmentons l'amplitude de l'onde (autant pour les ondes cnoïdales que pour les ondes de Stokes), ou plus nous augmentons les variations dans son profil, en passant par exemple d'une onde cnoïdale avec un plateau central à une sinusoïde, plus notre facteur intégrant modifié est supérieur au facteur intégrant classique. Cela va avec le fait que plus ce dernier perd de son efficacité, plus notre nouvelle méthode est, elle, efficace. En quelque sorte, nous sommes capables de rattraper les difficultés numériques que subit le facteur

intégrant classique.

Comme nous l'avons étudié, le facteur intégrant modifié peut être développé à plusieurs ordres, la limite venant soit des dérivées accessibles via le Dense Output, soit des erreurs numériques que nous imposons au système en voulant le forcer avec un polynôme d'ordre de plus en plus grand. Il serait donc intéressant de tester cette méthode sur des équations dont les solutions se rapprocheraient au maximum d'un polynôme. Cela devrait nous permettre d'être encore plus performant que pour le cas des vagues étudiées ici.

Afin de prolonger cette étude, nous pouvons aussi envisager d'autres axes de recherche. Le premier pourrait être de chercher une autre définition pour le polynôme approximant la partie non linéaire, dans l'origine du développement de Taylor, non plus autour du temps initial de la boucle de calcul, mais par exemple au temps intermédiaire (ce qui aurait un coût de calcul supplémentaire, mais il faudrait vérifier si les gains ne surpasseraient pas ce surplus de temps). Une deuxième idée est qu'au lieu d'utiliser une approximation de Taylor, nous pourrions, par exemple, regarder l'effet d'une approximation de Tchebychev. Comme l'objectif fixé pour ce travail n'était pas de comparer toutes les méthodes existantes entre elles, mais de montrer qu'une modification d'une des méthodes existantes peut permettre de fortement améliorer son efficacité, il pourrait être possible de regarder de plus près les différences entre notre modification et d'autres méthodes, telles que les *ETD*, du moment qu'il en existe avec les différentes méthodes emboîtées de Runge-Kutta que nous utilisons, afin de pouvoir avoir un technique de pas de temps adaptatif efficace. Enfin, il pourrait être intéressant de tester notre approche pour des types de schémas d'avance temporelle autre que ceux de Runge-Kutta, comme, par exemple, les méthodes de type Rosenbrock qui sont déjà utilisées pour certains intégrateurs exponentiels.

Annexe A

Détails pour le modèle *High-Order Spectral*

Pour approcher ce modèle plus en détails et ses différentes caractéristiques, il est possible de se référer aux thèses [8, 28] qui ont tout simplement été reprises pour cette annexe.

1 Equations reformulées en potentiel de surface

Le point de départ de ce modèle est de formuler en quantités surfaciques les Conditions de Surface Libre en $z = \eta(x, y, t)$,

$$\begin{aligned} \frac{\partial \eta}{\partial t} &= \frac{\partial \eta}{\partial z} - \nabla\phi \cdot \nabla\eta \\ \frac{\partial \phi}{\partial t} &= -g\eta - \frac{1}{2}\left|\widetilde{\nabla}\phi\right|^2 \end{aligned} \tag{A.1}$$

avec $\widetilde{\nabla}$ représentant le gradient volumique. Ces conditions, initialement formulées pour l'élévation de surface libre η et le potentiel des vitesses ϕ, sont exprimées en fonction de η et du potentiel de surface ϕ^s défini par

$$\phi^s(x, y, t) = \phi(x, y, z = \eta(x, y, t), t) \tag{A.2}$$

Nous obtenons ainsi les nouvelles *CSL* en $z = \eta(x, y, t)$

$$\begin{cases} \dfrac{\partial \eta}{\partial t} = & \left(1 + |\nabla\eta|^2\right)\dfrac{\partial \phi}{\partial z} - \nabla\phi^s \cdot \nabla\eta \\ \dfrac{\partial \phi^s}{\partial t} = & -g\eta - \dfrac{1}{2}|\nabla\phi^s|^2 + \dfrac{1}{2}\left(1 + |\nabla\eta|^2\right)\left(\dfrac{\partial \phi}{\partial z}\right)^2 \end{cases} \tag{A.3}$$

Ces deux équations nous permettent d'avancer en temps les quantités qui nous intéressent, à savoir $\eta(x,y,t)$ et $\phi^s(x,y,t)$. En effet, en supposant ces quantités connues à l'instant t, il est possible de les évaluer à l'instant $t+\Delta t$ en utilisant (A.3). On s'aperçoit alors que la seule inconnue restant dans le système d'équations précédent est la vitesse verticale sur la surface libre

$$W(x,y,t) = \frac{\partial \phi}{\partial z}(x,y,z=\eta(x,y,t),t) \tag{A.4}$$

De plus, il est à noter qu'il s'agit de la seule quantité volumique restante. Cette inconnue ne pouvant pas être obtenue de manière immédiate (à cause de la condition sur la surface libre), elle va être évaluée par un processus itératif d'ordre élevé correspondant au modèle *HOS*. Ce développement en quantités surfaciques permet donc une simplification très importante du problème initial, exprimé dans un certain volume de fluide. Ainsi, la résolution du problème se passe au niveau de la surface libre, ce qui fait que nous avons pu supprimer une dimension d'espace, la composante verticale z. Le développement en quantités spectrales s'effectue donc sur les composantes η et ϕ^s

$$\begin{aligned} \eta(x,y,t) &= \sum_i \sum_j A_{ij}^{\eta}(t)\Psi_{ij}(x,y,t) \\ \phi^s(x,y,t) &= \sum_i \sum_j A_{ij}^{\phi^s}(t)\Psi_{ij}(x,y,t) \end{aligned} \tag{A.5}$$

avec $k_{ij} = (k_{xi}, k_{yj})$ les nombres d'onde. Pour information, les fonctions de bases utilisées pour un milieu ouvert sont $\Psi_{ij}(x,y,t) = \exp(\mathrm{i}k_{x_i}x)\exp(\mathrm{i}k_{y_i}y)$, avec $(i,j) \in [-\infty,+\infty]$. Dans le cas d'un bassin de houle, nous avons $\Psi_{ij}(x,y,t) = \cos(k_{x_i}x)\cos(k_{y_i}y)$, avec $(i,j) \in [0,+\infty]$.

2 Calcul de la vitesse verticale

La première étape du calcul de la vitesse verticale $W(x,y,t)$ consiste à décomposer en série de puissance de η le potentiel ϕ

$$\phi(x,y,\eta,t) = \sum_{m=1}^{\infty} \phi^{(m)}(x,y,\eta,t) \tag{A.6}$$

Cette somme est ensuite tronquée à une valeur finie, appelée ordre *HOS* et notée M. Nous effectuons alors un développement de Taylor de chaque potentiel $\phi^{(m)}$ autour de $z=0$

$$\phi^{(m)}(x,y,\eta,t) = \sum_{n=0}^{\infty} \frac{\eta^n}{n!} \frac{\partial^n \phi^{(m)}}{\partial z^n}(x,y,0,t) \tag{A.7}$$

En combinant ces deux développements nous obtenons

$$\begin{aligned} \phi^s(x,y,t) &= \phi(x,y,\eta,t) \\ &= \phi^{(1)}(x,y,0,t) + \eta(x,y,t)\frac{\partial \phi^{(1)}}{\partial z}(x,y,0,t) + \cdots \\ &+ \phi^{(2)}(x,y,0,t) + \eta(x,y,t)\frac{\partial \phi^{(2)}}{\partial z}(x,y,0,t) + \cdots \end{aligned} \tag{A.8}$$

et nous regroupons alors chaque ordre de η comme

$$\begin{aligned}
\phi^{(1)}(x,y,0,t) &= \phi^s(x,y,t) \\
\phi^{(2)}(x,y,0,t) &= -\eta(x,y,t)\frac{\partial\phi^{(1)}}{\partial z}(x,y,0,t) \\
\phi^{(3)}(x,y,0,t) &= -\eta(x,y,t)\frac{\partial\phi^{(2)}}{\partial z}(x,y,0,t) - \frac{\eta^2}{2!}(x,y,t)\frac{\partial^2\phi^{(1)}}{\partial^2 z}(x,y,0,t) \\
\cdots &= \cdots \\
\phi^{(m)}(x,y,0,t) &= -\sum_{k=1}^{m-1}\frac{\eta^k}{k!}(x,y,t)\frac{\partial^k\phi^{(m-k)}}{\partial^k z}(x,y,0,t)
\end{aligned} \tag{A.9}$$

A noter que si nous avions plutôt $\phi^{(1)}(x,y,0,t) = \phi^s(x,y,t) + \delta(x,y,t)$, où $\delta(x,y,t)$ serait une petite perturbation, nous obtiendrions des erreurs non négligeables puisque cette valeur serait élevée à la puissance M.
Nous obtenons ainsi un système triangulaire, chaque ligne représentant une itération menée dans le calcul. Celles-ci sont évaluées sur des surfaces simples (en $z = 0$) et peuvent donc être calculées facilement par une méthode spectrale. Chaque ordre est décomposé sur des fonctions de base comme explicité précédemment en (II.1).

$$\phi^{(m)}(x,y,z,t) = \sum_i\sum_j A_{ij}^m(t)\Psi_{ij}(x,y))\frac{\cosh\left(k_{ij}\left[z+h\right]\right)}{\cosh\left(k_{ij}h\right)} \tag{A.10}$$

Les amplitudes modales sont obtenues pour chaque ordre $A_{ij}(t)$ à partir du système précédent (A.9). Nous pouvons aussi former un autre système afin de calculer la vitesse verticale W recherchée, celle-ci étant décomposée en série de puissances

$$W(x,y,t) = \sum_{m=1}^{\infty} W^{(m)}(x,y,t) \tag{A.11}$$

Nous en déduisons alors

$$\begin{aligned}
W^{(1)}(x,y,t) &= \frac{\partial\phi^{(1)}}{\partial z}(x,y,0,t) \\
W^{(2)}(x,y,t) &= \frac{\partial\phi^{(2)}}{\partial z}(x,y,0,t) + \eta(x,y,t)\frac{\partial^2\phi^{(1)}}{\partial^2 z}(x,y,0,t) \\
W^{(3)}(x,y,t) &= \frac{\partial\phi^{(3)}}{\partial z}(x,y,0,t) + \eta(x,y,t)\frac{\partial^2\phi^{(2)}}{\partial^2 z}(x,y,0,t) + \frac{\eta^2}{2!}(x,y,t)\frac{\partial^3\phi^{(1)}}{\partial^3 z}(x,y,0,t) \\
\cdots &= \cdots \\
W^{(m)}(x,y,t) &= \sum_{k=1}^{m-1}\frac{\eta^k}{k!}(x,y,t)\frac{\partial^{k+1}\phi^{(m-k)}}{\partial^{k+1} z}(x,y,0,t)
\end{aligned} \tag{A.12}$$

Ayant obtenu les amplitudes modales $A_{ij}(t)$ avec le système d'équations (A.9), nous avons accès aux différents ordres de la vitesse verticale $W^{(m)}$ et nous pouvons ainsi la calculer à l'ordre M voulu

$$W_M(x,y,\eta,t) = \sum_{m=1}^{M} W^{(m)}(x,y,\eta,t) \tag{A.13}$$

Nous pouvons noter que cette expression est la même dans les deux articles de Dommermuth et Yue [26] et de West *et al.* [80]. Ces derniers proposent un traitement du terme en W homogène

en ordre dans les *CSL*, que nous adoptons dans notre étude. Dans la Condition Cinématique de Surface Libre nous écrivons ainsi

$$\left(1+|\nabla\eta|^2\right)W \simeq W_M + |\nabla\eta|^2 W_{M-2} \tag{A.14}$$

alors que la formulation de Dommermuth et Yue est

$$\left(1+|\nabla\eta|^2\right)W \simeq W_M + |\nabla\eta|^2 W_M \tag{A.15}$$

De même, pour le terme $\left(1+|\nabla\eta|^2\right)W^2$ dans la Condition Dynamique de Surface Libre nous avons

$$\left(1+|\nabla\eta|^2\right)W^2 \simeq \sum_{p+q\leq M} W_p W_q + |\nabla\eta|^2 \sum_{p+q\leq M-2} W_p W_q \simeq \left(W^2\right)_M + |\nabla\eta|^2 \left(W^2\right)_{M-2} \tag{A.16}$$

Bibliographie

[1] J. H. Verner (Website). http://people.math.sfu.ca/~jverner. - *Cité 1 fois : page 40* -

[2] Matlab (Math Software). http://www.mathworks.fr/products/matlab. - *Cité 1 fois : page 76* -

[3] Thomas Brooke Benjamin, Jerry Lloyd Bona, and John Joseph Mahony. Model equations for long waves in nonlinear dispersive systems. *Philosophical Transactions of the Royal Society of London. Series A*, 272 :47–78, 1972. - *Cité 1 fois : page 16* -

[4] Thomas Brooke Benjamin and J. E. Feir. The disintegration of wave trains on deep water part 1. Theory. *Journal of Fluid Mechanics*, 27(03) :417–430, 1967. - *Cité 1 fois : page 110* -

[5] Håvard Berland. Exponential integrators. In *Department of Mathematical Sciences, NTNU, Norway. Oct 18, University of Central Florida*, 2005. - *Cité 1 fois : page 45* -

[6] Gregory Beylkin, James M. Keiser, and Lev Vozovoi. A new class of time discretization schemes for the solution of nonlinear PDEs. *J. Comput. Phys.*, 147(2) :362–387, 1998. - *Cité 2 fois : pages 46 et 47* -

[7] Przemyslaw Bogacki and Lawrence F. Shampine. A 3(2) pair of Runge-Kutta formulas. *Applied Mathematics Letters*, 2(4) :321 – 325, 1989. - *Cité 1 fois : page 40* -

[8] Félicien Bonnefoy. *Modélisation expérimentale et numérique des états de mer complexes*. PhD thesis, Université de Nantes, 2005. - *Cité 1 fois : page 121* -

[9] Joseph Valentin Boussinesq. Théorie de l'intumescence liquide, appelée onde solitaire ou de translation, se propageant dans un canal rectangulaire. *Comptes rendus de l'Académie des Sciences*, 72 :755–759, 1871. - *Cité 1 fois : page 12* -

[10] Joseph Valentin Boussinesq. Essai sur la théorie des eaux courantes. *Mémoires présentés par divers savants, l'Acad. des Sci. Inst. Nat. France, XXIII*, pages 1–680, 1877. - *Cité 1 fois : page 13* -

[11] John P. Boyd. Chebyshev and fourier spectral methods. pages xvi+668, 2001. - *Cité 1 fois : page 49* -

[12] J. C. Butcher. On the convergence of numerical solutions to ordinary differential equations. *Math. Comp.*, 20 :1–10, 1966. - *Cité 1 fois : page 58* -

[13] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations, Second edition*. John Wiley and Sons, Ltd, 2008. - *Cité 1 fois : page 37* -

[14] Claudio Canuto, M. Y. Haussaini, Alfio Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics*. Springer Series in Computational Physics, Springer-Verlag, Berlin, 1988. - *Cité 1 fois : page 27* -

[15] John D. Carter and Rodrigo Cienfuegos. The kinematics and stability of solitary and cnoidal wave solutions of the serre equations. *European Journal of Mechanics - B/Fluids*, 30(3) :259 – 268, 2011. - *Cité 1 fois : page 20* -

[16] J. Certaine. The solution of ordinary differential equations with large time constant. *Math. methods for digital comput.*, page 128–132, 1960. - *Cité 1 fois : page 46* -

[17] F. Ceschino. Modification de la longueur du pas dans l'intégration numérique par les méthodes à pas liés. *Chiffres*, 2 :101–106, 1961. - *Cité 1 fois : page 41* -

[18] A. Chabchoub, N. P.Hoffmann, and N. Akhmediev. Rogue wave observation in a water wave tank. *Phys. Rev. Lett.*, 106 :204502, May 2011. - *Cité 1 fois : page 17* -

[19] Didier Clamond. Steady finite-amplitude waves on a horizontal seabed of arbitrary depth. *Journal of Fluid Mechanics*, 398(-1) :45–60, 1999. - *Cité 1 fois : page 12* -

[20] Didier Clamond. Cnoidal-type surface waves in deep water. *Journal of Fluid Mechanics*, 489(-1) :101–120, 2003. - *Cité 1 fois : page 12* -

[21] Didier Clamond and Denys Dutykh. Practical use of variational principles for modeling water waves. *Physica D : Nonlinear Phenomena*, 241(1) :25 – 36, 2012. - *Cité 1 fois : page 18* -

[22] Didier Clamond and John Grue. A fast method for fully nonlinear water-wave computations. *Journal of Fluid Mechanics*, 447(-1) :337–355, 2001. - *Cité 1 fois : page 28* -

[23] S.M. Cox and P.C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2) :430 – 455, 2002. - *Cité 5 fois : pages 46, 47, 51, 57, et 69* -

[24] M. W. Dingemans and A. K. Otta. *Advances In Coastal And Ocean Engineering*. World Scientific Publishing CO. Pte. Ltd., 2001. - *Cité 1 fois : page 17* -

[25] Maarten W. Dingemans. *Water Wave Propagation Over Uneven Bottoms : Non-linear wave propagation*. World Scientific, 1997. - *Cité 1 fois : page 16* -

[26] Douglas G. Dommermuth and D. K. P. Yue. A high-order spectral method for the study of nonlinear gravity waves. *J. Fluid Mech.*, 184(-1) :267–288, 1987. - *Cité 2 fois : pages 21 et 123* -

[27] J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6 :19–26, 1980. - *Cité 1 fois : page 38* -

[28] Guillaume Ducrozet. *Modélisation des processus non-linéaires de génération et de propagation d'états de mer par une approche spectrale*. PhD thesis, Université de Nantes, 2007. - *Cité 1 fois : page 121* -

[29] K.B. Dysthe and K. Trulsen. Note on breather type solutions of the NLS as model for freak waves. *Phys. Scripta*, 82 :48–52, 1999. *- Cité 1 fois : page 110 -*

[30] W. S. Edwards, Laurette S. Tuckerman, Richard A. Friesner, and D. C. Sorensen. Krylov methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 110(1) :82–102, 1994. *- Cité 1 fois : page 47 -*

[31] J. D. Fenton. Nonlinear wave theories. *The Sea, Ocean Engineering Science, Eds. B. Le Méhauté and D. M. Hanes, Wiley Interscience, New York*, 9(A) :3–25, 1990. *- Cité 2 fois : pages 21 et 34 -*

[32] E. Fermi, J. Pasta, and S. Ulam. Studies of nonlinear problems. *Los Alamos Scientific Laboratory Report*, (No. LA-1940), 1955. *- Cité 1 fois : page 110 -*

[33] Bengt Fornberg. A practical guide to pseudospectral methods. *Cambridge Univ. Press, Cambridge, Royaume-Uni*, 1995. *- Cité 2 fois : pages 23 et 26 -*

[34] Armin Friedli. Verallgemeinerte Runge-Kutta Verfahren zur Löesung steifer Differentialgleichungssysteme. In R. Bulirsch, R. Grigorieff, and J. Schröder, editors, *Numerical Treatment of Differential Equations*, volume 631 of *Lecture Notes in Mathematics*, pages 35–50. Springer Berlin / Heidelberg, 1978. 10.1007/BFb0067462. *- Cité 2 fois : pages 46 et 51 -*

[35] Richard A. Friesner, Laurette S. Tuckerman, Bright C. Dornblaser, and Thomas V. Russo. A method for exponential propagation of large systems of stiff nonlinear differential equations. *Journal of Scientific Computing*, 4 :327–354, 1989. 10.1007/BF01060992. *- Cité 1 fois : page 47 -*

[36] Dorian Fructus, Didier Clamond, John Grue, and Øyvind Kristiansen. An efficient model for three-dimensional surface wave simulations : Part I : Free space problems. *Journal of Computational Physics*, 205(2) :665 – 685, 2005. *- Cité 1 fois : page 28 -*

[37] Richar M. Goodwin. *A Growth Cycle*, volume 19. In C. H. Feinstein, ed. Socialism, Capitalism and Economic Growth. Essays presented to Maurice Dobb. Cambridge : Cambridge University Press, 1967. *- Cité 1 fois : page 8 -*

[38] A. E. Green, N. Laws, and P. M. Naghdi. On the theory of water waves. *Proceedings of the Royal Society of London*, A(338) :43–55, 1974. *- Cité 1 fois : page 18 -*

[39] Eugene P. Gross. Structure of a quantized vortex in boson systems. *Il Nuovo Cimento Series 10*, 20(3) :454–477, 1961. *- Cité 1 fois : page 17 -*

[40] Kjell Gustafsson, Michael Lundh, and Gustaf Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28(2) :270–287, 1988. *- Cité 1 fois : page 42 -*

[41] Ernst Hairer, Syvert Paul Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I : Nonstiff Problems.* Springer Series in Computational Mathematics, 2000. *- Cité 7 fois : pages 36, 38, 41, 42, 63, 64, et 65 -*

[42] Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II : Stiff and Differential-Algebraic Problems.* Springer Series in Computational Mathematics, 2002. *- Cité 2 fois : pages 7 et 42 -*

[43] K. Henderson, D. Peregrine, and J. Dold. Unsteady water wave modulations : Fully nonlinear solutions and comparison with the nonlinear Schrödinger equation. *Wave Motion*, 29 :341–361, 1999. *- Cité 2 fois : pages 17 et 110 -*

[44] Marlis Hochbruck and Christian Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 34(5) :pp. 1911–1925, 1997. *- Cité 1 fois : page 51 -*

[45] Marlis Hochbruck and Alexander Ostermann. Exponential runge-kutta methods for parabolic problems. *Appl. Numer. Math.*, 53(2) :323–339, 2005. *- Cité 2 fois : pages 51 et 53 -*

[46] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19 :209–286, 2010. *- Cité 1 fois : page 45 -*

[47] R. Holland. Finite-difference time-domain (FDTD) analysis of magnetic diffusion. *IEEE Trans. Elect. Comp.*, 36(1) :32–39. *- Cité 1 fois : page 47 -*

[48] Diederik Johannes Korteweg and Gustav de Vries. On the change of form of long waves advancing in a rectangular canal and on a new type of long stationary waves. *Philosophical Magazine*, 39(39) :422–443, 1895. *- Cité 2 fois : pages 12 et 13 -*

[49] S. Krogstad. *Topics in numerical Lie group integration.* PhD thesis, University of Bergen, 2004. *- Cité 1 fois : page 55 -*

[50] S. Krogstad. Generalized integrating factor methods for stiff PDEs. *Journal of Computational Physics*, 203(1) :72 – 88, 2005. *- Cité 5 fois : pages 46, 55, 57, 68, et 83 -*

[51] J. Douglas Lawson. Generalized Runge-Kutta processes for stable systems with large Lipschitz constants. *SIAM Journal on Numerical Analysis*, 4(3) :pp. 372–380, 1967. *- Cité 3 fois : pages 46, 49, et 51 -*

[52] Elliott H. Lieb and Werner Liniger. Exact analysis of an interacting bose gas. I. The general solution and the ground state. *Physical Review*, 130 :1605–1616, 1963. *- Cité 1 fois : page 17 -*

[53] M. Lighthill. Contributions to the theory of waves in nonlinear dispersive systems. *J. Inst. Math. Appl.*, 1 :269–306, 1965. *- Cité 1 fois : page 110 -*

[54] Alfred James Lotka. Elements of physical biology. *Baltimore : Williams and Wilkins Co*, 1925. *- Cité 1 fois : page 8 -*

[55] Borislav V. Minchev. *Exponential integrators for semilinear problems.* PhD thesis, University of Bergen, Norway, 2004. *- Cité 3 fois : pages 45, 55, et 57 -*

[56] Borislav V. Minchev and Will Wright. A review of exponential integrators for first order semi-linear problems. In *Tech. report 2/05, Department of Mathematics, NTNU*, 2005. *- Cité 4 fois : pages 45, 46, 57, et 58 -*

[57] Robert M. Miura, Clifford S. Gardner, and Martin D. Kruskal. Korteweg-de Vries equation and generalizations. II. Existence of conservation laws and constants of motions. *J. Math. Phys.*, 9(8) :1204–1209, 1968. *- Cité 1 fois : page 20 -*

[58] Hans Munthe-Kaas. High order Runge-Kutta methods on manifolds. *APPL. NUMER. MATH*, 29 :115–127, 1999. *- Cité 2 fois : pages 46 et 55 -*

[59] Syvert Paul Nørsett. An A-stable modification of the Adams-Bashforth methods. In J. Morris, editor, *Conference on the Numerical Solution of Differential Equations*, volume 109 of *Lecture Notes in Mathematics*, pages 214–219. Springer Berlin / Heidelberg, 1969. 10.1007/BFb0060031. *- Cité 1 fois : page 46 -*

[60] Steven Alan Orszag. Comparison of pseudospectral and spectral approximation. *Studies in Applied Mathematics*, 51 :253–259, 1972. *- Cité 1 fois : page 24 -*

[61] A. Osborne, M. Onorato, and M. Serio. The nonlinear dynamics of rogue waves and holes in deep-water gravity wave train. *Phys. Rev. A*, 275 :386–393, 2000. *- Cité 1 fois : page 110 -*

[62] John Scott Russell. Report on waves. *Report of the Fourteenth Meeting of the British Association for the Advancement of Science Held at York in September 1844*, John Murray, London :311–390, 1845. *- Cité 1 fois : page 12 -*

[63] Mki. Sato, K. Kawabata, and J.E. Hansen. A fast invariant imbedding method for multiple scattering calculations and an application to equivalent widths of CO2 lines on venus. *Astrophys. J.*, 216 :947–962, 1977. *- Cité 1 fois : page 47 -*

[64] Fernando J. Seabra-Santos, Dominique P. Renouard, and A. M. Temperville. Numerical and experimental study of the transformation of a solitary wave over a shelf or isolated obstacle. *Journal of Fluid Mechanics*, 176(-1) :117–134, 1987. *- Cité 1 fois : page 18 -*

[65] F. Serre. Contribution à l'étude des écoulements permanents et variables dans les canaux. *Houille Blanche*, pages 374–388 and 830–872, 1953. *- Cité 1 fois : page 18 -*

[66] Lawrence F. Shampine. Interpolation for runge-kutta methods. *SIAM J. Numer. Anal.*, 22 :1014–1027, 1985. *- Cité 1 fois : page 64 -*

[67] Lawrence F. Shampine and Herman A. Watts. The art of writing a Runge-Kutta code. II. *Applied Mathematics and Computation*, 5(2) :93 – 121, 1979. *- Cité 1 fois : page 42 -*

[68] Trond Steihaug and Arne Wolfbrandt. An attempt to avoid exact jacobian and nonlinear equations in the numerical solution of stiff differential equations. *Mathematics of Computation*, 33(146) :pp. 521–534, 1979. *- Cité 1 fois : page 46 -*

[69] K. Strehmel and R. Weiner. Behandlung steifer anfangswertprobleme gewöhnlicher differentialgleichungen mit adaptiven Runge-Kutta-methoden. *Computing*, 29(2) :153–165, 1982. *- Cité 1 fois : page 46 -*

[70] C. H. Su and C. S. Gardner. Korteweg-de Vries equation and generalizations. III. derivation of the Korteweg-de Vries equation and Burgers equation. *J. Math. Phys.*, 10 :536–539, 1969. *- Cité 1 fois : page 18 -*

[71] C. Sulem and P. L. Sulem. *The nonlinear Schrödinger equation. Self-focusing and wave collapse.* Applied Mathematical Sciences, Springer-Verlag, New York, 1999. *- Cité 1 fois : page 17 -*

[72] Allen Taflove. *Computational electrodynamics : The finite-difference time-domain model.* Artech House, 1995. *- Cité 1 fois : page 47 -*

[73] David Le Touzé. *Méthodes spectrales pour la modélisation non-linéaire d'écoulements à surface libre instationnaires.* PhD thesis, Université de Nantes, 2003. *- Cité 1 fois : page 24 -*

[74] Ch. Tsitouras. Runge-Kutta interpolants for high precision computations. *Numerical Algorithms*, 44 :291–307, 2007. *- Cité 1 fois : page 66 -*

[75] Pieter Jacobus Van der Houwen and J. G. Verwer. *Generalized linear multistep methods I. Development of algorithms with zero-parasitic roots.* Mathematisch Centrum, Amsterdam, 1974. *- Cité 2 fois : pages 46 et 47 -*

[76] J. H. Verner. Explicit Runge-Kutta methods with estimates of the local truncation error. *Siam Journal on Numerical Analysis*, 15, 1978. *- Cité 1 fois : page 40 -*

[77] J. H. Verner. Differentiable interpolants for high-order Runge-Kutta methods. *Siam Journal on Numerical Analysis*, 30, 1993. *- Cité 1 fois : page 66 -*

[78] J. G. Verwer. On generalized linear multistep methods with zero-parasitic roots and an adaptive principal root. *Numerische Mathematik*, 27 :143–155, 1977. 10.1007/BF01396634. *- Cité 1 fois : page 47 -*

[79] Vito Volterra. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. R. Accad. Naz. dei Lincei. Ser. VI*, 2, 1926. *- Cité 1 fois : page 8 -*

[80] Bruce J. West, Keith Allen Brueckner, Ralph S. Janda, David Michael Milder, and Robert L. Milton. A new numerical method for surface hydrodynamics. *J. Geophys. Res.*, 92(C11) :11803–11824, 1987. *- Cité 2 fois : pages 21 et 123 -*

[81] Will Wright. Exponential integrators : A review. In *Department of Mathematical Sciences, NTNU, Norway. ANZIAM 05, January 30 - February 4*, 2005. *- Cité 1 fois : page 45 -*

[82] Theodore Yaotsu Wu. A unified theory for modeling water waves. volume 37 of *Advances in Applied Mechanics*, pages 1 – 88. Elsevier, 2001. *- Cité 1 fois : page 18 -*

[83] Vladimir Evgen'evich Zakharov. Instability of waves in nonlinear dispersive media. *J. Exp. Theor. Phys.*, 51 :1107–1114, 1966. *- Cité 1 fois : page 110 -*

Résumé

Pour réaliser des simulations précises aux temps longs pour des vagues non linéaires, il faut faire appel à des algorithmes d'évolution temporelle précis. En particulier, la combinaison d'un pas de temps adaptatif avec un *facteur intégrant* est connue pour être très efficace. Nous proposons une modification de cette technique. Le principe consiste à soustraire un certain polynôme à une *EDP*. Puis, comme pour le facteur intégrant, nous faisons un changement de variable pour retirer la partie linéaire. Mais nous espérons retirer quelque chose de plus afin de rendre l'*EDP* moins raide pour les calculs numériques. Le polynôme choisi est une expansion de Taylor autour du temps initial de la solution. Afin de calculer les différentes dérivées nécessaires, nous utilisons le *Dense Output* qui donne la possibilité d'approximer les dérivées de la solution à tout temps. Une fois le facteur intégrant modifié appliqué, nous faisons appel à une avance temporelle classique afin de résoudre l'équation d'évolution. Il a été considéré plusieurs schémas de Runge-Kutta avec pas de temps adaptatif. Nous avons tiré avantage des méthodes emboîtées, afin de ne pas calculer de nouvelles fonctions et perdre du temps de calcul, en utilisant uniquement des données déjà calculées durant l'évolution temporelle. Les résultats numériques montrent que l'efficacité de notre méthode varie selon les cas. Par exemple, nous avons vérifié que plus le profil de l'onde est pentue, plus notre méthode est efficace. Pour le modèle de vagues non linéaires le plus compliqué à notre disposition, le modèle *HOS*, nous avons pu réduire le nombre de pas de temps de calcul jusqu'à près de 30 % avec un schéma de Runge-Kutta de Dormand-Prince et jusqu'à plus de 99 % pour un schéma de Bogacki-Shampine.

Mots-clés : *Intégrateur exponentiel, Facteur intégrant, Vagues non linéaires, Equations de Serre, Modèle HOS, Pas de temps adaptatif, Dense Ouput, Réduction de temps de calculs.*

Abstract

Efficient time stepping algorithms are crucial for accurate long time simulations of nonlinear waves. In particular, adaptive time stepping combined with an *integrating factor* are known to be very effective. We propose a modification of the existing technique. The trick consists in subtracting a certain-order polynomial to a *PDE*. Then, like for the integrating factor, a change of variables is performed to remove the linear part. But, here, we hope to remove something more to make the *PDE* less stiff to numerical resolution. The polynomial is chosen as a Taylor expansion around the initial time of the solution. In order to calculate the different derivatives, we use a *dense output* which gives a possibility to approximate the derivatives of the solution at any time. The modified integrating factor being applied, a classical time-stepping method can be used to solve the remaining equation. We focus on various Runge-Kutta schemes with a varying step size. We take advantage of embedded methods and use an evolved adaptive step control. We do not need to calculate new functions and loose time of calculation only by using already estimated values during the temporal evolution. Numerical tests show that the actual efficiency of the method varies along cases. For example, we verified that steeper waves profiles give rise to better behaviour of the method. For fully nonlinear water wave simulations with the *HOS* model, we can save up to 30% of total time steps with a Dormand-Prince Runge-Kutta scheme and we can save up to 99% with the Bogacki-Shampine scheme.

Keywords : *Exponential integrator, Integrating Factor, Nonlinear waves, Serre's equations, HOS model, Adaptive step size, Dense Output, Reducing computation time.*